FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Wireless Sensor Network for Forest Fire Detection

João Teixeira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Aníbal Ferreira

Co-orientador: Prof. Pedro Pinto

25th June, 2017

Abstract

Portugal, like many countries all over the world, is affected every year by fires that consume and destroy thousands of hectares of forest territory and endanger people lives, homes and properties. Despite the investment and evolution in fire fighting techniques and equipment, it is common for fires to achieve a greater proportion then they should, due to late detection.

Considering the evolution on sensing and monitoring systems, allied to the development of radio communication technologies such as the IEEE 802.15.4 standard, a wireless communications protocol that specifies low-data-rate for personal networks, it is now possible to create a system capable of monitoring and detecting a fire occurrence, in a remote area, as soon as it starts.

The project under study in this dissertation aims at creating such a system and including innovative features such as a variable reading rate mode when the probability for fire occurrence is higher. For that purpose an implementation with ten MTM-CM5000 motes, IEEE 802.15.4 standard compliant devices, will be developed and tested. Contiki Os, together with Cooja simulator were the main software used to code the devices. RPL protocol, as supported by Contiki OS, was the routing protocol implemented. A web application was also developed to provide alerts when a fire occurrence is most likely to occur and present the collected data from the sensor network as well as the network connections and routes in real time. The development for both the wireless sensor network and test procedures are included in this dissertation.

Both simulation and experimental tests results that are presented indicate that the implemented system functionalities are consistent with the desired behaviour.

Resumo

Portugal, tal como muitos outros países, é afetado todos os anos por fogos que consomem e destroem milhares de hectares de território florestal e colocam em perigo pessoas e bens. Apesar do investimento e desenvolvimentos em técnicas e equipamentos de combate aos incêndios, é comum uma ocorrência de incêndio tomar proporções mais elevadas que o suposto, devido a uma deteção tardia.

Tendo em conta as evoluções nos campos e sistemas de sensorização e monitorização, aliadas ao desenvolvimento de tecnologias de comunicação sem fios, como por exemplo o standard IEEE 802.15.4, um protocolo de comunicação sem fios especifico para redes com baixas taxas de transmissão, é hoje possível desenvolver um sistema capaz de monitorizar uma área remota e detetar um possível evento de incêndio, assim que este começa.

O projeto em estudo nesta dissertação tem como objetivo criar um sistema como o acima referido, introduzindo algumas funções inovadoras tais como a possibilidade de variar a rate de leitura de um nó sensor assim que este deteta uma possível situação de perigo. Para esse efeito, será desenvolvida uma implementação com dez dispositivos MTM-CM5000, dispositivos esses compatíveis com o protocolo IEEE 802.15.4.

O Contiki OS, juntamente com o simulador Cooja, foram o software mais utilizados durante o desenvolvimento do projeto. O protocolo de routing RPL, bem implementado no Contiki OS, foi selecionado como o protocolo de *routing* a aplicar na rede.

Foi também desenvolvida uma aplicação web para produzir alertas quando algum possível evento de perigo é detetado e para apresentação dos dados recolhidos na rede de sensores. A descrição dos procedimentos utilizados durante a fase de desenvolvimento, tanto para a rede como para a parte web são enunciados nesta dissertação. Tanto os resultados de teste obtidos em simulação como os executados experimentalmente com os dispositivos indicam que a funcionalidades são consistentes com o esperado.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ICNF	Instituto da Conservação da Natureza e das Florestas
WSN	Wireless Sensor Network
PAN	Personal Area Network
IEEE	Institute of Electrical and Electronics Engineers
WPAN	Wireless Personal Area Network
SN	Sensor Node
P2P	Peer-to-Peer
LP-WPAN	Low-Power Wireless Personal Area Network
RFD	Reduced Function Device
FFD	Full Function Device
MAC	Media Access Control
GTS	Guaranteed Time Slot
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
CPU	Central Processing Unit
CCA	Clear Channel Assessment
UWB	Ultra Wide Band
SAP	Service Access Point
RED	Received Energy Detection
NWK	Network Layer
APS	Application Support Layer
ZDO	ZigBee Device Object
REP	Random Equal Probability Forwarding
SZ	Source Zigzag Forwarding
DZ	Destination ZigZag Forwarding
SFN	Single Frequency Network
OS	Operating System
NesC	Nested C
MCF	Minimum Cost path Forwarding
EU	European Union
IOT	Internet Of Things
DODAG	Destination Oriented Acyclic Graphs
DAG	Directed Acyclic Graph
DIO	DODAG Information Object
LLN	Low-power and Lossy Network
AODV	Ad Hoc On-Demand Distance Vector
RREQ	Route Request Command
RREP	Route Reply Packet
ETX	Expected Transmission Count
RSSI	Received Signal Strength Indicator
GUI	Graphical User Interface
GSM	Global System for Mobile Communications
GPRS	General Packet Radio Service

Chapter 1

Introduction

1.1 Motivation

Portugal is affected, every year, by fires that destroy thousands of hectares of forest territory and also endanger people lives, goods and property.

Even though there is an increase on fighting means and fire-fighters' qualifications, sometimes, these are not enough to extinguish the fire as quickly as possible due to the delay in the response times. In general, a late detection allows a fire to achieve greater proportions making it harder to extinguish it.

One element that could improve the fire-fighters' performance and other fire extinguishing related authorities would be to detect and release a fire alert as soon as possible. The shortest detection and alert time allows for an earlier action from the authorities, drastically improving their odds on the fire extinguish task.

According to provisional data for 2015 and previous years, made available by ICNF, it can be verified that most of the burnt area corresponds to forest territory. In fact, due to the remote location of some of these areas, when the alert is given, the fire has already reached a proportion that impairs its fast extinction by the fire-fighter teams and other authorities.

The main purpose of this work is precisely the development of a semi-autonomous system for fire detection in remote territory, by monitoring on site several fire-related indicators. A wireless sensor network will be deployed and will be constantly monitoring the target area. It will be possible to detect, alert and retrieve information about the state of an ongoing fire in a simpler, faster and more effective way.

1.2 Context

The evolution on sensor systems, together with the development of new protocols and wireless communication techniques allows us today to apply in real life some concepts that before were not possible. Some of the previous limitations were the power management, the communication

network and sensor node implementation, the costs and some other factors that increased these type of projects complexity and implementation difficulty.

With the IEEE 802.15.4 protocol, that refers to very low power and low cost wireless communication, it is now possible to develop and implement a WSN whose purpose is to help in detecting and solving real-life problems. One such issue, recurrent in Portugal and the rest of the world, is the destruction of thousands of hectares of forest territory by wildfires, every year.

This project addresses the above challenges, as a WSN that is capable of monitoring in real-time a widespread area of forest in an autonomous way, allied with a system that allows the collection, processing and access of data directly from the site, will allow the authorities to greatly reduce the response time in a wildfire occurrence, and provide critical information about the wildfire characteristics allowing for better personnel and equipment management. These improvements will come with the earlier wildfire alert and data collection about the fire characteristics, especially when implemented in remote or hard to access areas.

1.3 Objectives

Being the study subject the creation of a wireless sensor network for fire detection, the aim is to idealize a system, that when deployed on a site will monitor certain physical indicators related to fire conflagration like temperature, humidity, luminosity and certain gases' presence such as Carbon Dioxide and Monoxide (CO2 and CO).

With the IEEE 802.15.4 protocol, that allows wireless communication with reduced power consumption, a WSN will be created for the collection and transmission of data from several sensor nodes. The ZigBee standard, or other with similar or better characteristics, will be used with this intent. The data should then be accessed and presented in a Web interface or page that will allow to visualize the state of the monitored area and emit fire alerts upon detection by any of the sensor nodes.

The main objectives are the successful development and implementation of a WSN in a test area with around 10 hectares, the configuration of a web interface and system that will allow for data management and retrieval, history recording, producing statistics, emit or activate alerts and alarms when needed and show in real-time the state of the WSN and monitored values.

Chapter 2

Wireless Sensor Networks Fundamentals

In this chapter the main objective is to make a review on the fundamentals of Wireless Sensor Networks. This review will allow for a better insight on a great variety of topics whose familiarity is required for the successful completion of the project in study. The main aspects to be focused on this chapter are the technological bases and knowledge necessary for a WSN deployment. Firstly the concept of WSNs will be studied, then an overview on the low power wireless standard IEEE 802.15.4 will be made. Some insights on the WSN routing topologies and protocols, sensor node installation and IEEE 802.15.4 compatible devices will also be approached.

2.1 WSN - Wireless Sensor Network Overview

2.1.1 WSN History

Like a lot of technological developments and advances, the concept of a wireless network with some autonomous elements or devices monitoring and analysing the surrounding environment in order to collect, process and analyse data, was driven mainly by military and defence organizations. In fact, the development of this technology was done in four main stages with military and defence organizations as the main boosters or supporters of this concept in the beginning, as stated in [10], [13], [1], [5] and shown in Figure 2.1.

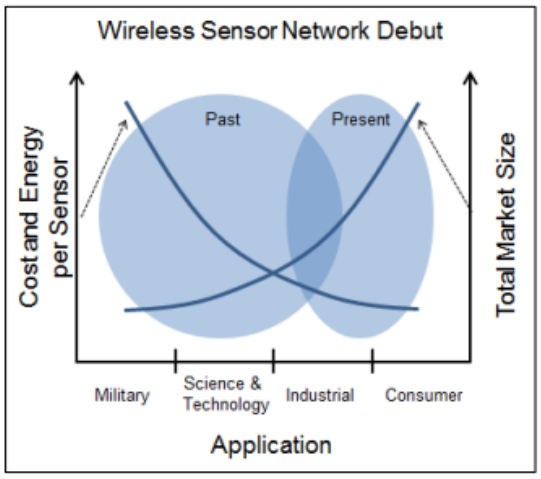


Figure 2.1: WSNs Gain Market Traction with Decrease in Sensor Costs [1].

2.1.1.1 1st Phase - During cold war (50's)

During the cold war, the need to detect underwater submarines resulted in the installation of a first remote monitoring system that was based on the detection of underwater sound waves. A vast amount of sensors was deployed on ocean floor and the SOSUS (*Sound Surveillance System*) was implemented. Since 1994 SOSUS is used to monitor oceanic events such as animal or seismic activity [13], [1].

2.1.1.2 2nd Phase - Advanced Defense Projects (80's)

The first modern research on WSN can be said to have begun around 1980 when DARPA (*Defence Advanced Research Projects*) initiated the *Distributed Sensor Networks* project [13]. The project was conceived with the aim of placing sensor nodes in a distributed, cooperative and low cost architecture that were autonomous and able to share the network resources. It was a really ambitious project considering that at this time the existence of personal computers or workstations was not yet a widespread reality. The processing of data was mainly done by minicomputers. The network was based on acoustic sensors and its proper operation was proven by the successful tracking of a low flying aircraft [13].

2.1.1.3 3rd Phase - Military Sensor Networks in the late 20th century

In the last two decades of the 20th century, the awareness of the potential and importance of WSN technologies became even more relevant to military defence organizations which lead to the increase in research and development of WSNs and related technologies. At this time, the technology for small sensors was not yet developed enough but the possibilities this technology brought in tracking performance, detection range and faster response times made WSN a crucial component of study and development [13]. The first main applications of this concept were limited

and based on bulky and expensive sensors and proprietary networking protocols, focusing mostly on performance and functionality [1].

2.1.1.4 4th Phase - Beginning of 21st century to today

Due to the advances in networking, processing, energy management and production techniques that significantly reduced the costs of building and implementing these WSN based systems, the main focus of WSN use begun to shift from military applications to more consumer directed and industrial related ones as shown in [1]. In fact, the lower cost of deployment and increasing functionality of WSN led to even more interest on the development and research on these networks and also led to some joint efforts between academia and industry in developing technologies and communication and network standards for WSN. From 1999 to 2002 the following initiatives took place [1]:

- 1999 - University of California at Berkeley PicoRadio program,
- 2000 - Micro Adaptive Multi-domain Power Aware Sensors program at MIT ,
- 2001 - NASA Sensor Webs,
- 2002 - ZigBee Alliance,
- 2002 - Center for Embedded Network Sensing.

These standards and other related developments tried to achieve a reduction in costs while trying to increase functionality and simplify development, management and maintenance tasks enabling high-volume deployment of WSNs. A major factor for today's WSNs implementation is also the sensor node size that, as shown in 2.1, has been greatly reduced to the point of reaching nano-technology based sensor nodes [12],[10],[13].

Although still developing, we can begin to see a more widespread use of WSNs in several fields, from small WSNs for consumers at home to massive deployments to monitor extensive areas of interest.

2.1.2 WSN Applications

Due to the potential in monitoring and alert possibilities of WSNs, the proliferation and expansion of this type of networks is currently increasing and spreading through several areas of study and real life applications. Due to the almost unlimited potential for WSN applications, the whole field spectrum is very difficult to define. Some of the most prominent fields where WSN are being applied today are:

- **Area monitoring** - Both for military and consumer use in intrusion or breach detection in the WSN deployment area allowing for better security of infrastructures or areas,

Table 2.1: Sensor Node Evolution [12], [13].

	1st Gen (80's 90's)	2nd Gen, Early 2000s	3rd Gen. Today (2010+)
Size	Shoe box	Book or smaller	Coin size or smaller, nano size
Weight	Kilograms	Grams	Milligrams or negligible
Node Architecture	Separated sensing, processing and communications	Integrated sensing, processing and communications	Fully integrated sensing, processing and communication
Protocol	Proprietary	Mainly propietary, Wi-Fi, ZigBee, WiMax, others	Standard: 802.15.4, Wi-Fi, ZigBee, WiMax, others
Topology	Point-to-Point, star, multihop	Client-Server, P2P	Fully P2P
Power Supply	Large batteries or line feed	AA batteries	AAA batteries, Solar/Wind generated, others
Life Span	Hours or Days	Days to weeks	Months to Years

- **Natural Disaster prevention and detection** - Sensor nodes are deployed in a study area and collect data from the surrounding environment, analysing certain physical values that allow authorities to detect a possible disaster before it occurs or right after it happens, reducing response time and life and property loss. There are ongoing and successfully implemented projects on detection of fire, floods, land slides and other natural disasters that may occur,

- **Healthcare** - The medical and healthcare applications are still developing. Patient related applications, consist on either implanted or wearable devices that will be in charge of the overall monitoring of patients and their health status by collecting relevant data. WSNs can also be deployed on a hospital allowing for better management of staff, patients and resources,

- **Environmental monitoring** - WSNs are used in this area mainly for monitoring water quality or air pollution,

- **Industrial monitoring** - Industries can implement WSNs for machine health or other production related monitoring, detecting machine faults, controlling production and many other factors that are relevant to the industry itself,

- **Smart Home/Office** - Sensor nodes deployed on a room or set of rooms can provide information about physical elements such as temperature, luminosity, noise, gas presence and/or other variables that impact the living or working conditions.

The history of WSN, has its relevance for the project under study since it provides a better understanding about how the technology evolved from the very first developments until recent days. By understanding the evolution of the technology under study, a better insight on what the

next possible developments for WSNs will be and what characteristics they will have improved in the near future is achieved.

2.2 WSN Main Characteristics

In this section, the structural elements, characteristics and concepts to have in mind in the design of a Wireless Sensor Network will be under focus. Further specifications and analysis of WSN constituting elements and technologies will be under study in the next sections.

2.2.1 Main characteristics / challenges

When idealizing and building a fully functional WSN there are some requirements or challenges that must be considered and analysed to ensure the correct working functionalities or tasks. According to [10], [14] and [5] the main characteristics that must be considered include the following.

- **Composition** - If the sensor network is composed of sensor nodes that have the same hardware characteristics such as CPU power, memory, composing sensors, communication module and others, we can say that the WSN is **Homogeneous**, on the other hand, if these specifications are not the same for all sensor nodes the network is then said to be **Heterogeneous**,
- **Reliability** - WSNs are based on wireless communication, and as such, they are vulnerable to problems like packet loss due to a lot of factors (dispersion, reflection, diffraction and other). Reliability of the network must be ensured in order to overcome these problems,
- **Power consumption** - The power efficiency, specially in remote places with no access to grid power, is one of the most crucial characteristics that must be analysed. Usually, in these types of deployments, and due to their size, the sensor nodes are powered by batteries and this limits their lifetime, working modes and functionality,
- **Node size** - For installation and production purposes, the size of the sensing unit or sensor nodes must be considered,
- **Mobility** - Mobility wise, a WSN may be **Stationary** or **Mobile**. In a stationary network, the deployment and installation of sensor nodes is set and remains unchanged. On the other hand, a mobile network refers to WSNs comprised of mobile sensor nodes whose position may change. Even in a stationary network, due to the probability of a sensor node breaking or malfunctioning the implementation of the WSN, despite being stationary, they may consider mobility requirements,
- **Privacy and Security** - From the very fundamental of WSN, wireless communication, a lot of potential security problems arise. Protecting the system and the transmitted data requires a greater concern in these networks since wireless channels are accessible both by legitimate and illegitimate users. In order to protect the entire system and WSN encryption of the data and traffic encoding must be considered amongst other security measures,
- **Organization** - A WSN may be characterized in two different categories: **Hierarchical** or **Non-Hierarchical** considering the role each sensor node performs. If the sensor nodes fulfil

the same role it can be said that the network is Non-Hierarchical but when there are sensor groups fulfilling different roles/tasks on different organizational levels then the network is said to have an Hierarchical organization,

- **Density** - This characteristic represents the density of node sensors in a certain measured area. A network may be considered **Irregular**, **Balanced**, **Dense** or **Scattered**. Depending on the way the sensor nodes are deployed in an area the network may be irregular if there is not an organized disposition of the sensing devices and balanced if there is an organized installation. Considering the number of sensors on a determined and limited area, the network can also be described as dense if there is a great number of sensor nodes in a limited area or scattered if that number is low,

- **Data acquisition** - The data acquisition may be performed in several ways. It may be **Continuous**, **Queried**, **Periodic** or **Reactive**,

  - **Continuous** - the data is continuously being collected and sent,
  - **Queried** - the data is collected and sent upon a query from the management system or user,
  - **Scheduled** - the data is collected and sent on a set of pre-defined time intervals,
  - **Reactive** - the data is collected and sent when a certain value or parameter exceeds a pre-defined threshold.

2.2.2 WSN Main Elements/Entities

Like any type of network, a WSN has a minimum set of required components, entities or elements for the correct implementation and working mode.

A basic sensor network must rely on a minimum number of basic elements or components that will later be studied in more detail. These basic elements or entities that must be considered are the *Sensor Nodes*, the *Gateway*, the *Network topology / Information Path*, the *Collected data* and the *Data management system(s)*. The sensor node specification will be dealt with later in this dissertation (Section 2.2.2.1). In general, it consists of the local device(s) with integrated sensors, CPU, memory, communication module and other components that allow the monitoring target physical values to be collected from the environment, processed and then forwarded in the network. The gateway is a device that will act as a sink node and to where all the collected data and values must be sent to so that the data can be processed and/or stored in a data base and then forwarded to the management system. The network topology and information path refers to or defines a virtual path of data flow created between sensor nodes, gateway and management system that allows for data forwarding from one end to the other of the whole monitoring system. The collected data is the environmental or physical parameters that the sensor nodes collect from each sensor. To this data the sensor node will add some extra information upon data transmission (unique id, location, battery status or others). The data management system is the system where

all the collected data is processed, stored and made available for visualization and interaction by a human user. Figure 2.2 illustrates and example of a WSN with a grid topology and its main composing elements.

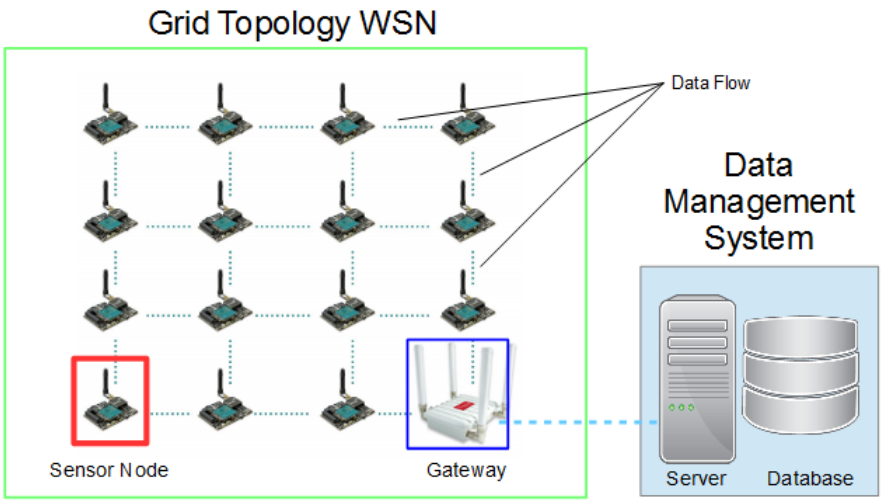


Figure 2.2: WSN main elements in grid topology.

2.2.2.1 Sensor Node

The sensor node, also known as mote, is responsible for data collection, monitoring and transmitting information comprised of the physical variables monitored on a determined area. The SN is considered to be the main component of a WSN, since the interconnection between them is what creates the WSN.

Some characteristics or considerations to have about the sensor nodes are [2]:

- SNs are densely deployed,
- SNs are prone to failures,
- The topoly of a WSN may change frequently,
- SN are limited in power, computational capacity and memory,
- SNs may not have global identifications because of the large number of sensors.

Hardware wise, a sensor node mainly consists of three main elements, CPU, communication module and sensors [12], [10], or it can have more elements like an actuator, a power generator a location finding system and others, as can be seen in Figure 2.3.

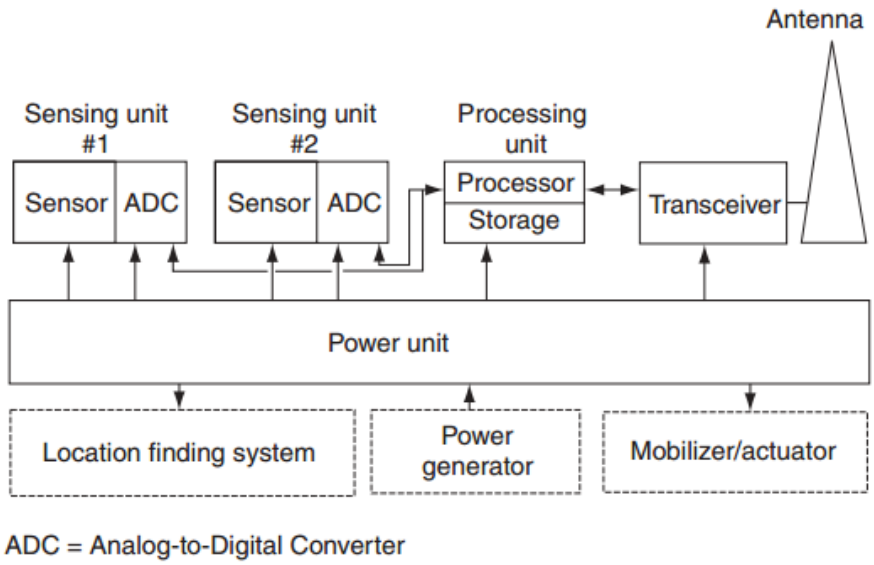


Figure 2.3: Typical Sensing Device [2].

For the project under study, the main expected elements that will make the sensor node are the following:

- **Sensor/ADC** - The sensor is the unit that captures a real-life physical value from the surrounding environment (temperature, humidity, gas concentration, and others) and translates them to electrical signals. These are then converted from analogue to digital signals by the ADC unit, for the CPU to process,
- **CPU** - The controller or CPU is a component that performs tasks, processes the data and controls the other components functionalities. Usually in SN this component is a microcontroller, due to power, size and cost of an SN,
- **Storage/Memory** - Usually it consists of flash memory when it is required in SN related devices. Usually there are two purposes for it, it may be considered program memory or user memory. The first is for programming the device and user memory may be necessary to store user, device or application data in the device,
- **Power Unit** - This unit is responsible for the storage and supply of energy to the various SN components. Usually a battery or multiple batteries are used,
- **Power Generator** - It consists of the unit or device responsible for charging the power unit. It may harvest solar, wind or be connect to a power line.

Identifying all the above WSN characteristics and composing elements is likely to be one of the most crucial reviews to be made when developing a wireless sensor network such as the one from this project. Without this understanding of the core composing elements and main network

characteristics for a wireless sensor network the development of a project including such a network can not be successful.

2.3 IEEE 802.15.4

2.3.1 Overview

The IEEE 802.15.4 Protocol is a wireless communications standard that specifies low-data-rate for personal wireless networks. This standard is fundamental for the deployment of Low-Power WPANs (LP-WPAN) that primarily require low power consumption and low cost of manufacture, due to its low data-rate proprieties and simplicity, IEEE 802.15.4 is the proper base solution for these types of networks.

Due to its role on low-power communications several other protocols/standards were developed with IEEE 802.2.15.4 as their core and will be analysed on the Alternatives section.

2.3.2 Network Model

The basic IEEE 802.15.4 standard consists of two main types of device and three types of elements each with a specific set of functions or roles within the network. As stated in [10], these two types of device are:

- **FFD** *Full Function Device* - A device that may assume one of three roles: **PAN coordinator**, **coordinator** or a **simple device**. This device may communicate with either RFDs or other FFDs,

- **RFD** *Reduced Function Device* - This is a simple device and is not allowed to assume the role of either a coordinator or PAN coordinator. This device is usually implemented with simpler and lower performance hardware and makes periodic communications to an FFD, the only device an RFD may communicate with.

A wireless network based on this standard is required to have at least a working FFD device and one or more RFD devices. The required FFD device is the PAN coordinator and it is the device that allows for the establishment and maintenance of an IEEE 802.15.4 network. The remaining RFD devices, with lower working capabilities and functions, collect and communicate the data to the coordinator [15].

Considering two different types of network topologies [15], [10], Star topology and Peer-to-Peer topology, Figure 2.4, have distinct requirements for their constituting elements:

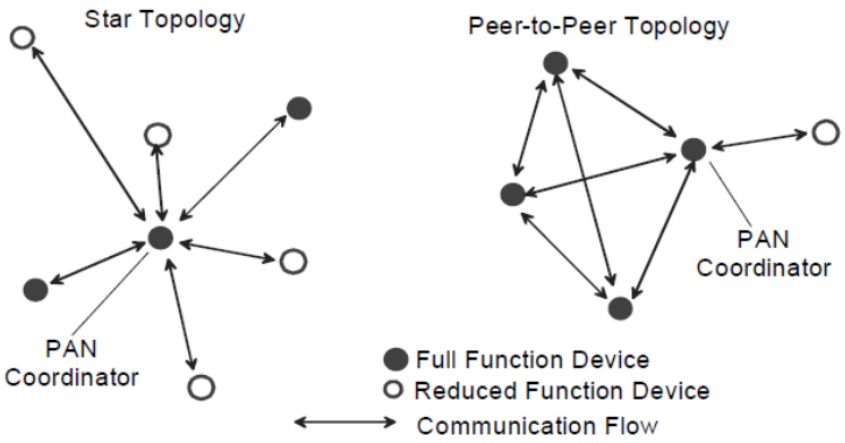


Figure 2.4: Star and Peer to Peer Topology [3].

We can see in Figure 2.4 that the Star topology has an FFD device as a PAN coordinator to which all the other RFD devices are connected and communicate with. On the other hand, a P2P topology is built with only or mainly FFD devices, allowing every node to communicate with each other. For environmental applications, [10], concludes that Peer-to-Peer based topologies are usually the best ones, for they allow to obtain data from the whole network and from each sensor node or device and allowing for a more fail-proof system, since a failure in a single sensor node will not include the risk of a set of sensor nodes failing, as in a Star topology could occur if one of the central FFD devices was at fault. The complexity of the network must also be considered, and in the case of a more complex and larger network, Peer-to-Peer is also the best solution [15].

2.3.3 Protocol Structure

In this subsection an analysis of the protocol structure of the IEEE 802.15.4 standard will be made mainly by addressing the Media Acess Control Layer, (*MAC*), and the Physical Layer,(*PHY*).

2.3.3.1 Physical Layer

The physical layer is defined as the first OSI model. It provides a connection between the MAC sublayer and the physical radio channel and must handle the following tasks [3]:

- Activation and deactivation of the radio transceiver,
- Energy detection (ED) within the current channel,
- Link quality indicator (LQI) for received packets,
- Clear channel assessment (CCA) for carrier sense multiple access with collision avoidance (CSMA-CA),
- Channel frequency selection,

- Data transmission and reception,
- Precision ranging for ultra-wide band (UWB) PHYs.

The physical layer also specifies a transmit power of at least 0.5 milliwatt, a receiver sensitivity of less than or equal to 85dbm in the 2.5GHz band [2], [4]. Other parameters are also specified according to the used frequency band. With a higher frequency band, the Data Rate and the number of channels increase, as shown in Figure 2.5.

	868 MHz	902-928 MHz	2.450 GHz
Data Rate	20 kbps	40 kbps	250kbps
# channels	1	10	16
TX Power	-3dBm	-3dBm	-3dBm
RX Sensitivity	-92dBm	-92dBm	-85dBm
Link Budget	89dB	89dB	82dB
Adjacent channel rejection	0dB	0dB	0dB
Alternate channel rejection	30dB	30dB	30dB

Figure 2.5: IEEE 802.15.4 PHY Parameters [4].

2.3.3.2 Media Access Control Layer

According to [3], the official standard definition, the MAC sub-layer is responsible for handling all accesses to the radio channel and performs the following tasks:

- Generating network beacons if the device is a coordinator,
- Synchronizing to network beacons,
- Supporting PAN association and disassociation,
- Supporting device security,
- Employing the CSMA-CA mechanism for channel access,
- Handling and maintaining the GTS mechanism,
- Providing a reliable link between two peer MAC entities.

The MAC frame format used for communications is expressed in Figure 2.6:

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
		Addressing fields						
MHR							MAC Payload	MFR

Figure 2.6: General MAC frame format [3].

Although the fields in MHR appear in a fixed order, the addressing fields may not be included in all frames.

The frame control field, as illustrated in 2.6, contains multiple sub sections. Some examples are the Frame type (Beacon, Data, Ack, MAC command or Reserved) or the Security Enabled section (one if frame is protected by MAC sublayer and zero otherwise).

The IEEE MAC protocol allows two working modes, a non-beacon enabled mode, in which nodes have access to the communication channel using classical, un-slotted CSMA-CA mechanisms and a beacon enabled mode, that in order to save energy resorts to using duty cycles with slotted CSMA-CA mechanisms [16]. The beacon enable mode is based on the concept of a superframe that is delimited by beacon frames sent by the coordinator. The superframe structure is divided in two main parts: active and inactive, as illustrated in Figure 2.7.

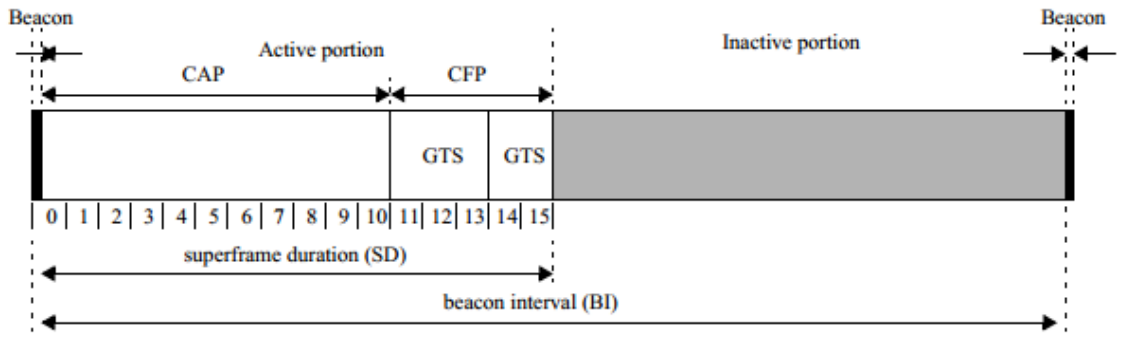


Figure 2.7: Example of superframe structure [3].

To provide a more personal use and simpler implementation of IEEE802.15.4 standard based networks and devices, several other technologies were developed with this standard as their core. One of the most widespread used wireless technologies of this kind is Zigbee, which has been developed as an open and global standard that operates on the IEEE 802.15.4 physical radio and MAC layer specifications defined above.

After concluding this analysis on the IEEE 802.15.4 protocol and its most defining operation characteristics, there is a better understanding about the capabilities of the network intended to be developed. Since IEEE 802.15.4 is at the core of this project, as it was the chosen wireless protocol to be used, it was necessary to better understand its network model and how it operates

in the Physical and Medium Access Layer to enable wireless communication between different devices. The next step is to understand how information flows in a network and what impact different techniques have in the network and its elements.

2.4 WSN Routing Strategies

Several routing strategies for the intended WSN implementation will be analysed and compared based on previous studies on the topic. Since a structured grid deployment of the sensor nodes, using a peer-to-peer topology is considered the most power and transmission efficient architecture [6], [14], [10], the routing techniques analysed in this section will be more focused on this type of topology, Figure 2.8 . This structure is used for covering large areas and networks [5]. Figure 2.8 shows the intended architecture design the WSN deployment should follow.

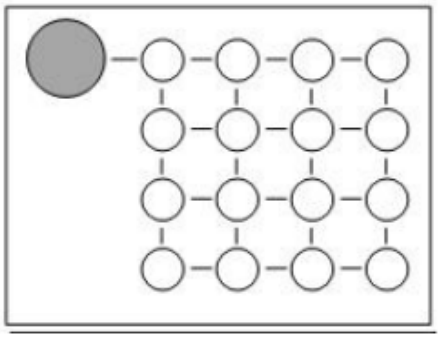


Figure 2.8: Example of grid topology with one sink node [5].

2.4.1 Routing Protocols Challenges

When considering WSN routing protocols, some requirements must be met [5]:

- **Number of transmissions** - The number of hops or transmissions a packet has to take to reach its destination must be as lower as possible. Every transmission uses a certain amount of energy. It is also important to reduce retransmissions of packets to achieve lower power consumptions,
- **Balanced use of nodes** - The load on the nodes for the routing should be balanced between all nodes. Overload in a node will cause it to consume more energy, several dead sensor nodes may result in a partial isolation of the network,
- **Delay** - For many applications, delay minimization is a top priority. For these delay sensitive applications it should be reduced to a minimum,
- **Balancing previous aspects** - The routing strategy must consider all previous aspects and relate them to the intended purpose of the WSN in study. Due to power and other limitations a trade-off between these aspects must be made accordingly.

2.4.2 Routing Strategies

Although the network topology is defined, the communication from the sensor nodes to the gateway must be done in the fastest and most effective way. For this purpose, the routing strategy is responsible for finding the best path for the information flow by applying some set of rules that the network elements must follow. Since the expected topology will be a grid, a study about grid related routing mechanisms in WSNs was done by [6] and will be the core of this section. There are several other routing technologies not under study in this dissertation, as each network varies in purpose, limitations and topologies and its own ideal routing strategy. For this project the main purpose is the efficiency and quality of the transmitted data and low power consumption.

In this study, typical source routing tactics are under study. The Sink Node will be located on the bottom left corner of the network, as shown in Figures 2.9 and 2.10. Considering the number of elements in the network, the grid maximum coordinates in x-axis and y-axis are *im* an *jm* respectively. The nodes in the origin axis transmit through that same axis and the rest may follow one of the following tactics [6].

2.4.2.1 Random Equal-Probability Forwarding

This tactic is fairly simple to implement and states that a forwarded data packet is sent through the left or down node with an equal 0.5 probability. When the packet arrives to a origin axis, either *x-axis* or *y-axis*, it is then forwarded in that same line, along the axis until it reaches the sink node.

2.4.2.2 Line Forwarding

This routing method is also quite simple, the packets are forwarded to either the *x-axis* or the *y-axis* with equal probability of 0.5. Like the one before, when the axis is reached the data packet travels along the axis until it reaches the origin.

2.4.2.3 Source Zigzag Forwarding

The data packets are forwarded to the left-down node via the left or the down node with equal probability. Like the ones before, this behaviour changes when the origin axis is reached and the packet is forwarded along the axis until it reaches the sink node, as can be seen in Figure 2.9.

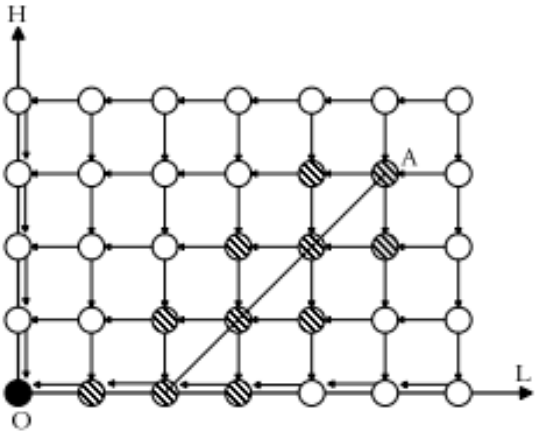


Figure 2.9: Source Zigzag routing [6].

2.4.2.4 Destination Zigzag Forwarding

Data packets are destined to the $\pi/4$ line that starts from the origin and travels along their axis until that line is reached. Then the packet will be forwarded to the left-down node as in SZ, through the left or down node with equal 0.5 probability. An example of a data packet being forwarded from the node A to the origin, Figure 2.10.

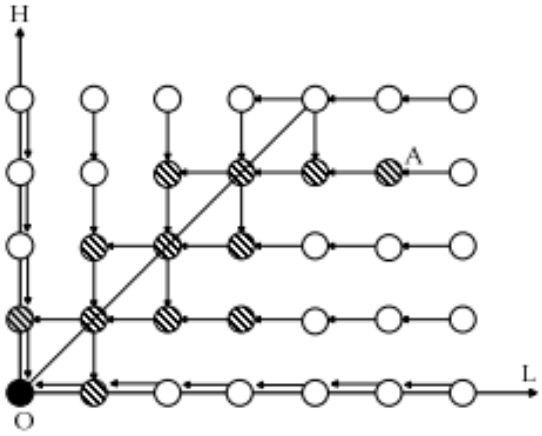


Figure 2.10: Destination Zigzag routing [6].

2.4.2.5 Balanced Forwarding

Probably the most complex one under study by [6]. In this technique, the nodes with the same distance to the sink node are clustered into the same line group. It is defined that the distance from a node (i,j) inserted into the S line group to the sink node, in number of hops, is calculated using the following expression: $N_{hops} = im + jm - 2 - s$ where $s = im + jm - i - j - 2$. It can be seen in Figure 2.11 that the grid is divided between an expansion $0 \leq s < jm - 1$, propagation $jm - 1 \leq s < im - 1$ and contraction $im - 1 \leq s < im + jm - 2$ region.

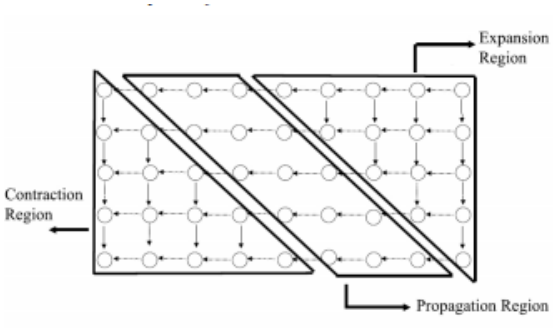


Figure 2.11: Balanced routing [6].

Label the n_t^s node as the *t-th* node starting from the end-point of line group s, and the p_t^s as the probability of transmitting from n_t^s to n_t^{s+1} we get the following equations, 2.1.

$$p_t^s = \begin{cases} \frac{s+2-t}{s+2}, & 0 \leq s < jm-1 \\ 1, & jm-1 \leq s < im-1 \\ \frac{im+jm-1-s-t}{im+jm-s-2}, & im-1 \leq s \leq im+jm-2 \end{cases} \tag{2.1}$$

2.4.3 Routing strategies results

The results on the study of each of these techniques is stated on the next following subsections.

2.4.3.1 Traffic Load

According to the results on traffic load testing, [6] observed that the nodes in the *x-axis* are heavily loaded for REP and SZ tactics. As for DZ tactic, the interior nodes are the ones that are more heavily loaded. For the In Line mechanism, results show that nodes at the same distance from the sink node in the *x-axis* and *y-axis* share similar loads but the the interior nodes are more heavily loaded. After all these considerations, it was concluded that Balanced mechanism was the best for traffic load.

2.4.3.2 Energy consumption

After the tests, it was shown that the overall energy consumption for the different transmission tactics was the Balanced mechanism. The SZ mechanism is the most energy consuming tactic, followed by REP. The tests had the same results for various operating modes, transmission and stand-by as illustrated in Figure 2.12.

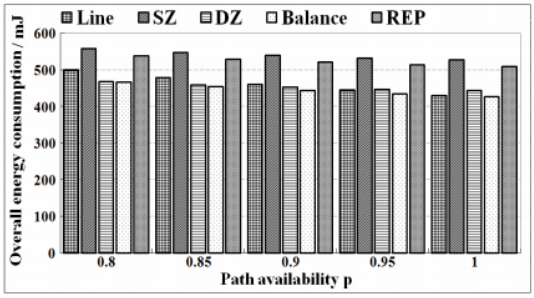


Figure 2.12: Overall Energy Consumption [6].

After the conclusion of this study it was shown that for grid topology, amongst the tested routing techniques, the Balanced method is the best one for the project under study.

Each one of the mentioned routing techniques, applied to wireless sensor networks in a grid topology, create different impact on the network elements, as shown by the mentioned studies, but are limited when it comes to recover from a faulty device since, by design, these techniques set the possible paths information may take.

Considering the application of the project under study, to develop a network whose devices would be deployed and subjected to adverse conditions in outdoor environments, there is a higher probability that at some point one or more devices will fail.

In order to mitigate the possibility of a total network failure due to a missing or faulty node, smart, self repairing and fault tolerant routing protocols, that also do not create too much overhead in the network while operating, some other protocols must be chosen. The next section will describe two of the most widely used routing protocols for WSNs that fall into the final described protocol category.

2.5 IEEE 802.15.4 Routing Protocols

This chapter will focus on two of the most used routing protocols in WSNs. An overview about the Ad Hoc On-Demand Distance Vector, AODV, and RPL, a routing protocol specially designed with Low-Power and Lossy Networks will be made.

2.5.1 Ad Hoc On-Demand Distance Vector - AODV

The Ad Hoc On-Demand Distance Vector, or AODV, routing protocol, was developed with mobile nodes in an ad hoc network in mind [17].

The main characteristics that AODV pretends to fulfil are to enable dynamic, self-starting, multi-hop communication between mobile nodes in an ad hoc network [17].

AODV's working fundamentals are based on the usage of tables in each node, containing information about neighbouring nodes and the next to hop to reach a certain destination node. The working methodology of this protocol is as follows: when a node wants to transmit to a certain

destination but does not yet have a route, a route discovery process takes place. The sender node will broadcast a route request command (RREQ) throughout the network. The RREQ contains the source network address, the destination network address and a path cost. The path cost field refers to the metric which is used to define the best routes, certain AODV based routing protocols may use different metrics to define the best path (EX. E-AODV utilizes the energy level of a node as the main metric).

As the RREQ command is propagated through the network, each node that re-sends this command updates the path cost field creating a temporary entry in the routing table. When the destination node finally receives de RREQ command, it compares the path cost with all the previously route request commands, defining the minimum cost one as the path to be used. The destination node then forwards a route reply packet (RREP) through the minimum cost path.

The route discovery process is then finished for this instance and the data transmission will then be possible from the source to the destination node.

This process, although introducing some extra network load will allow for a best fault tolerance behavior, since the routes are not necessarily static and when an intermediate node goes offline for some reason the network will automatically go around the issue by discovering a new best route.

2.5.2 RPL

As mentioned before, RPL is a protocol designed with the LLNs as its main target. The constrains that are typical of this type of network created the need for its development. These networks are often characterized by limited memory, energy and processing capabilities. The supported traffic flows by RPL are "point-to-point (between devices inside the LLN), point-to-multipoint (from a central control point to a subset of devices inside the LLN), and multipoint-to-point (from devices inside the LLN towards a central control point)" [18]. RPL adopts mechanisms that "facilitate forwarding data and minimizing routing complexity" [19], resulting in a reduction of memory requirements and signalling overheads. In terms of organization, RPL topology is organized in DAGs that are subdivided into one or more DODAGS. Each sink in a network has one DODAG [19]

In order to identify and maintain a determined topology, RPL bases its operation in four different values [18], [19]:

. **RPLInstanceID** - Responsible for identifying one or more DODAGS, Destination Oriented Acyclic Graphs,

. **DODAGID** - unique for each DODAG,

. **DODAGVersionNumber** - incremented when a DODAG root reconstructs a DODAG,

. **Rank** - used to identify the position of a specific node in relation to the DODAG root.

The rank of a node is determined by the Objective function. By default, RPL determines ranks based on ETX (Expected Transmission Count) metric. In terms of operation, RPL works in the following way [20]:

. **Upward Routing:**

- DAG information object (DIO) messages are broadcast periodically for discovering, forming and maintaining a DODAG tree,
- DIO includes:

  DODAG ID,

  Rank,

  ETX.
- Each node advertises its ETX to the sink node,

. **Downward Routing:**

- Destination Advertisement Option (DAO) messages can be sent by leaf nodes to advertise the paths to root nodes,
- As DAO travels upward through the DODAG to towards the root node, each node appends its unique ID when it passes through them.

This routing protocol allows for dynamic routing and fault tolerance and is the default protocol implemented on Contiki Os network stack.

As RPL is specifically designed for networks such as the one proposed for development, it is the routing protocol that best applies to be used in this project. The reviewed protocols have similar capabilities and when compared to the previously mentioned routing techniques, in 2.4, both of them are more appropriate to be utilised in a network such as the one proposed due to their fault tolerance capabilities introduced by dynamic routing techniques. Having the basic concepts for the network theory, the next step is to review the actual conditions and methods required to deploy the intended developed system on a determined area for monitoring. For that purpose, the next section reviews these and other issues.

2.6 WSN Topologies

In this section the method used to install the sensor nodes will be under study, considering communication efficiency, installation costs and other factors.

As mentioned before, the distribution and number of the sensors on the test or monitored area will affect the behaviour and the topology of the network. The number of sensors is considered of great importance since it must provide coverage to all the targeted area but it can not be an infinite number, since every sensor node or device added will also add interference to the network [21]. Besides the number of sensors, the means and organization used for their installation also reflects on the network operability and communication efficiency [14]. The last considerations we must make on this topic are also related to the terrain topology in the monitored area since obstacles such as walls, trees, vegetation and changes on terrain level may also introduce some noise or interference on communications.

Considering that this project field of study is the implementation of a WSN on a remote and mainly forest terrain with relatively low vegetation, the considerations, tests, results and studies of [14] will be applied to this wildfire detection system installation parameters since they study that exactly same terrain topology.

The installation process to adopt can be sectioned in several important steps like cost of installation and communication efficiency.

2.6.1 Installation Costs

When compared to a cable network, the WSNs' installation costs are relatively low since there is almost no need for infrastructure creation and the installation is mainly done by just placing the sensor nodes on their designated places. The costs of installation are not linearly related to the number of sensor nodes since the logistic and resources needed may differ greatly when increasing their number [14]. Also, certain parts of the monitored terrain may have conditions that will lower or raise the costs by making the access to the installation place harder, taking more time, effort and human labour, therefore increasing the costs.

2.6.2 Number of sensor nodes

When considering the number of sensor nodes to install we must consider the following equation [22]:

$$N = \frac{2.A.\pi}{r^2\sqrt{27}} \tag{2.2}$$

where **N** is the number of sensor nodes to calculate, **A** is the area that is going to be monitored and **r** is the effective sensing radius of the sensor node. Although this equation provides some help and a starting point on deciding the minimum number of sensors to cover a complete area, it does not consider the terrain and surrounding environment properties [14], and as such, only after testing the network, will the total number of required sensor nodes be determined.

2.6.3 Installation topology

After several tests, specially the ones focussing on the more relevant terrain topology for this project, Mediterranean forest with up to 50cm of vegetation, in [14], it was concluded that by installing the sensor nodes in an organized grid topology where the sensor nodes are distributed evenly across the terrain, one by one, was the most effective way to achieve the best communication quality and efficiency.

Another mentioned factor that greatly impacts the quality of the connection in this type of terrain is the correct orientation of the sensor node antennas, in an upwards orientation the quality is significantly greater, from around 75% connectivity to 100% from changing the antennas in a downward direction to the correct one [14].

This section aimed to define the guidelines for a WSN deployment in a real outdoor environment, considering the area to be covered and the number of sensor nodes to be used to adequately monitor it, their disposition in the terrain and the relation between the procedure and costs of installation of all the sensor nodes. The following section will be a market study that will provide a better knowledge about the existent solutions in the market to acquire or develop the mentioned sensor nodes.

2.7 WSN Devices and Modules

In this section the main objective is to find some *off-the-shelf* devices that are already available in the market and may be required for this project. After some research, a lot of devices and pre-existing communication modules and sensors are already available for purchase.

During the next subsections pre-existing IEEE 802.15.4 compliant modules and other crucial hardware that are already commercially available will be described and compared according to the most relevant characteristics for the project under study. The focus will be on the performance and range, power consumption, transmission rate, cost, implementation complexity or other relevant factors.

2.7.1 Motes / Sensor Nodes

When considering the implementation of a WSN, the sensor nodes are the core elements that must be studied. They act as a processing unit, as transmitters and receivers of collected data from the sensors or sensor boards enabling the creation of a mesh network by exchanging or propagating information.

A mote, together with the sensing devices, creates the Sensor Node. The main elements required for a fully functional Sensor Node are the microprocessor, a communication module, memory, one or more power sources and the sensors.

There are eveeral 802.15.4 or Zigbee oriented solutions. One of them was mentioned in 3.1, the Waspmote. Some other widely used devices for development or research in IOT projects are motes from the Telosb Family, Micaz, Zoolertia and many others. Also, on the market, may also be found some clones that were developed with the mentioned ones at their core.

2.7.1.1 Libelium's Waspmote

The Libelium's Waspmote, which architecture was described in subsection 3.1 in more detail, has to its advantage the fact that it has been already deployed on a wildfire-detection WSN. The general characteristics of this mote are:

. Microcontroller: ATmega1281

. Frequency: 14.7456 MHz

. SRAM: 8KB

. EEPROM: 4KB

. FLASH: 128KB

. SD Card: 2GB

. Weight: 20gr

. Dimensions: 73.5 x 51 x 13 mm

. Temperature Range: [-10 ^{o}C, +65 ^{o}C]

. Clock: RTC (32KHz)

Energy consumption wise, Waspmote's characteristics are as follows:

. ON - 15mA

. Sleep - 55μA

. Deep Sleep - 55μA

. Hibernate - 0.7μA

2.7.1.2 TelosB / Tmote Sky

TelosB, TPR2400, is an open source, low power consumption, IEEE 802.15.4 compliant device that was "developed and published to the research community by UC Berkeley", [7].

The main characteristics of this device are, [7]:

. IEEE 802.15.4/ZigBee compliant RF transceiver

. 2.4 to 2.4835 GHz, a globally compatible ISM band

. 250 kbps data rate

. Integrated onboard antenna

. 8 MHz TI MSP430 microcontroller with 10kB RAM

. 1MB external flash for data logging

. Optional sensor suite including integrated light, temperature and humidity sensor

. Features built-in battery socket for 2 AA batteries

. Module Current draw Active mode - 1.8 mA

. Module Current draw Active mode - 5.1μA

. RF Transceiver Current draw Receive mode - 23mA

. RF Transceiver Current draw Idle mode - 21μA

. RF Transceiver Current draw Sleep mode - 1μA

This device is widely used for WSN research purposes due to the easy of use and programming. To add to its features it must also be said that this device is compatible with both TinyOs and Contiki Os. The basic schematic for this device ant its components can be seen in Figure 2.13.

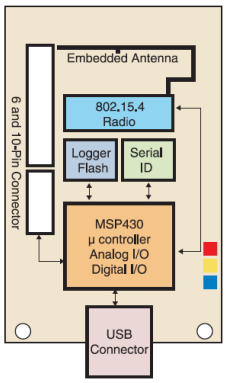


Figure 2.13: TPR2400 CA Block Diagram [7].

2.7.1.3 MicaZ

MicaZ, MPR2400, is also an IEEE 802.15.4 compliant mote that is specifically designed for sensor networks, [8]. It is based on the low power microcontroller ATmega128L. It features a 51 pin connector that supports UART, I2C, SPI and Analog Inputs interfaces, allowing to be versatile in connecting to multiple sensors or other hardware. This device main characteristics are, [8]:

. IEEE 802.15.4 compliant RF transceiver

. 2.4 to 2.48 GHz, a globally compatible ISM band

. 250 kbps data rate

. Supported by MoteWorks^{TM} wireless sensor network platform for reliable, ad-hoc mesh networking

. Plug and play with MEMSIC's sensor boards, data acquisition boards, gateways, and software

. CPU Current draw Active mode - 8 mA

. CPU Current draw Active mode - $< 15\mu A$

. RF Transceiver Current draw Receive mode - 19.7mA

. RF Transceiver Current draw Idle mode - $20\mu A$

. RF Transceiver Current draw Sleep mode - $1\mu A$

The block diagram for MicaZ mote can be seen in Figure 2.14.

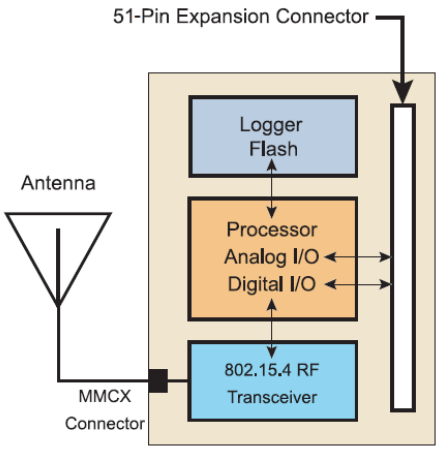


Figure 2.14: TPR2400 CA Block Diagram, [8].

2.7.2 Wireless communication modules

On wireless communication modules there are several manufacturers and products available for WSN implementation besides the previously mentioned ones that are built-in with some of the motes. Figure 2.15 shows some of the available ZigBee compliant modules in the market manufactured by Digi that are licensed for use in the EU. The modules may be used with a wide range of motes.

	 XBee ZigBee Embedded ZigBee modules provide OEMs with a simple way to integrate mesh technology into their application VIEW PRODUCT ▸	 XBee 802.15.4 Low-cost, easy-to-deploy modules provide critical end-point connectivity to devices and sensors VIEW PRODUCT ▸	 XBee DigiMesh 2.4 Wireless Mesh Networking RF Module VIEW PRODUCT ▸	 XBee-PRO 868 Long-Range RF Module for Europe VIEW PRODUCT ▸
Form Factor	Through-hole, Surface Mount	Through-hole	Through-hole	Throughhole
Transmit Power	6.3 mW (+8 dBm) / 63 mW (+18 dBm)	1 mW (+0 dBm) / 63 mW (+18 dBm); 10 mW (+10 dBm) International	1 mW (+0 dBm) / 63 mW (+18 dBm); 10 mW (+10 dBm) International	1 mW (0 dBm) to 315 mW (+25 dBm)
Certified Regions	US, CA, EU, AU, BR, JP	US, CA, EU, AU, BR, JP	US, CA, EU, AU, BR, JP	Europe
Sleep Current	< 1 uA	< 10 uA	< 50 uA	-
Receive Sensitivity	-102 dBm	-100 dBm	-100 dBm	-
Encryption	128-bit AES	128-bit AES	128-bit AES	-
Wireless Frequency	2.4 GHz (ISM)	2.4 GHz (ISM)	2.4 GHz (ISM)	-
Transmit Current	120 mA @ 18 dBm	215 mA @ 18 dBm	250 mA @ 18 dBm	-
Receive Current	31 mA	55 mA	55 mA	-
Networking Protocol	ZigBee®	Point-to-Point, Point-to-Multipoint	DigiMesh	-
Range (LoS) with high gain antenna	Up to 2 miles	Up to 1 mile	Up to 1 mile	-
Spread Spectrum	-	-	-	DSSS
Networking Topologies	-	-	-	Multipoint
Outdoor Line of Site Range	-	-	-	25 miles / 40 km
Protocols	-	-	-	Proprietary
Data Rate	-	-	-	24 Kbps (limited to 10% duty cycle)
USB Ports	-	-	-	868 MHz

Figure 2.15: Digi Xbee EU licensed modules example and specifications

2.7.3 Sensors

In the sensor market a lot of different elements may be found. They can go from really low cost sensors with lower performance characteristics and sensibilities with a great number of existing sensors of each type.

Depending on the application, a lot of different sensors may be acquired or built. For the Waspmotes, the company Libelium provides a pre-built board for gas detection which includes these and other sensors. This sensor board, in Figure 2.16, even without knowing its cost, may be a good solution since these sensors were already implemented in real life applications. The issue is the fact that those extra sensors may add to the board cost and power consumption making it non-viable for use in this project.

Figure 2.16: Libelium Gas Board.

In the Telosb and Micaz family, motes or clones already bring some built-in sensors as mentioned above. The most common sensors to be present in these type of devices are temperature, humidity and luminosity.

This market study provides a better insight on available components required when designing a WSN. Without this study, the capabilities of each available device on market would be unknown, which could determine the development of the project. When choosing each component or full pre-built devices there are some considerations to be made about them such as OS compatibility, cost, transmission power and range, energy consumption. After this review, an increase in knowledge about what hardware is available to choose from before developing this project.

Chapter 3

Literature Review

In this section, some previously developed systems related to the detection of forest fires and environmental monitoring using wireless networks will be under consideration. There are some developed systems for forest fire detection besides the ones under study in this section, but most of the systems developed are not WSN based solutions and are mostly image relying systems that use cameras on site or, in most cases, satellite imagery. These systems are not considered to be fully real-time detection systems since image processing and data processing is more time consuming. Considering the purpose of the project at hands, the following projects provide the best background information and study methodologies we could find for the development of this project.

3.1 SISVIA Vigilancia y Seguimiento Ambiental

Libelium is a WSN specialized Spanish company that provides WSN related technologies and devices, and has performed several joint case studies in the deployment of WSN applications for multiple purposes, [23]: agriculture applications, smart cities, radiation monitoring, industrial control, disaster detection such as wildfires and many others.

One of the most important case studies to consider in this dissertation is a joint effort between two spanish companies, DIMAP and Libelium. Under the name SISVIA Vigilancia y Seguimiento Ambiental, DIMAP developed and deployed a fire detection system using Libelium WSN technologies.

The proposed system was deployed in Galizia, Spain, and aimed for a 210 hectares area coverage, [24].

The main devices Libelium provided for this project development are the *Waspmotes*, the *Sensor boards*, *Wireless interfaces* and the *Meshlium*.

- **Waspmote** - the sensor node main processing unit and its core. It is the central piece of the WSN. The sensor board and wireless interface will be connected to the waspmote.

- **Sensor Board** - the board, available from Libelium, used to mount the enormous variety of sensors they provide. This sensor board will connect to the waspmote and will collect the surrounding environment parameters.
- **Wireless Interfaces** - This module is necessary for the communication between waspmotes and the network. Libelium provides a different variety of interfaces, each with its own characteristics. The ones in study will be the ones compatible with the IEEE 802.15.4/ ZigBee stack.
- **Meshlium** - This device will act as the network coordinator and gateway. According to libelium "*Meshlium is a Linux router which works as the Gateway of the Waspmote Sensor Networks. It can contain 6 different radio interfaces: WiFi 2.4GHz, WiFi 5GHz, 3G/GPRS, Bluetooth, XBee and LoRa.*", [25].

The intended deployed system can be observed in Figure 3.1.

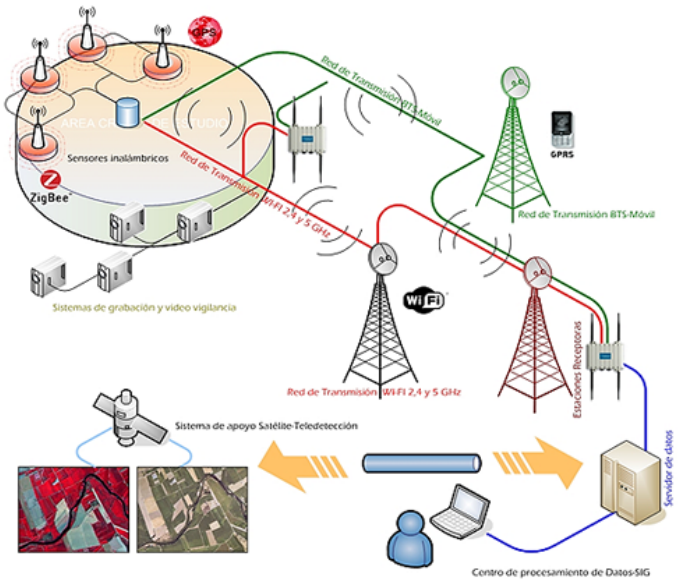


Figure 3.1: SISVIA Vigilancia y Seguimiento Ambiental system.

The Waspmote is the core product Libelium provides for WSN solutions as it is the device that acts as the brain of the whole sensor node. Libelium waspmotes are focused on low power consumption and their main features are, [9]:

1. $Microcontroller : ATmega1281$
2. $Frequency : 14.7456MHz$
3. $SRAM : 8KB$
4. $EEPROM : 4KB$
5. $FLASH : 128KB$

6. $SDCard : 2GB$
7. $Weight : 20gr$
8. $Dimensions : 73.5x51x13mm$
9. $TemperatureRange : [-10^oC, +65^oC]$
10. $Clock : RTC(32KHz)$

Energy wise, this device has 4 different power modes: ON, Sleep, Deep Sleep and Hibernate. Power consumption values refered in [9], are the following:

1. $ON : 15mA$
2. $Sleep : 55\mu A$
3. $DeepSleep : 55\mu A$
4. $Hibernate : 0.07\mu A$

The waspmote is prepared for interconnecting not only with the sensor board and the wireless module but with other available modules as well, GPS, GSM amongst others.

Figure 3.2 shows a waspmote with different modules already attached.

Figure 3.2: Wasmote and Meshlium Grid WSN [9].

Figures 3.3 and 3.4 represent the board and connectors of the waspmote.

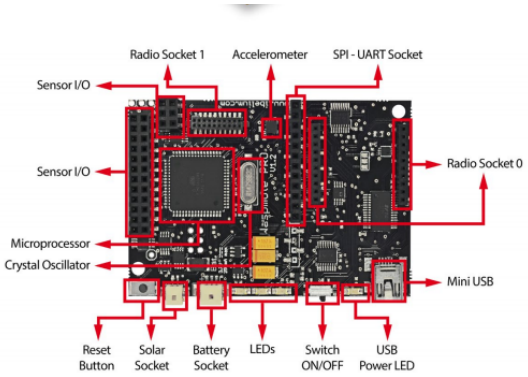


Figure 3.3: Wasmote top [9].

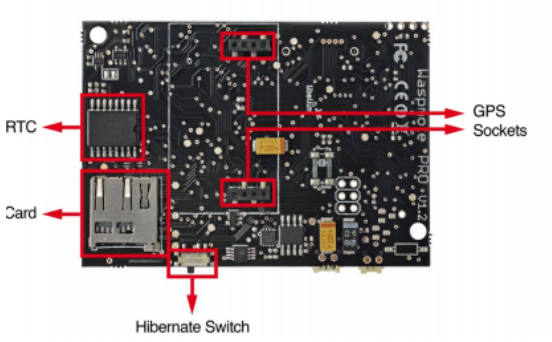


Figure 3.4: Wasmote Bottom [9].

For this project, 90 waspmotes were used to monitor four physical factors, temperature, relative humidity, carbon monoxide and carbon dioxide. For this purpose, Libelium's gas board which includes these and many other sensors were attached to each of the waspmotes. Each sensor power may be adjusted since each one has a separate power connection.

Regarding the communication modules, Libelium provides several fully compatible modules that are shown in Table 3.1.

Table 3.1: Available Libelium communication modules

Model	Protocol	Frequency	TX power	Sensitivity	Range*
XBee-802.15.4	802.15.4	2.4 GHz	1 mW	-92 dB	500 m
XBee-802.15.4-Pro	802.15.4	2.4GHz	63 mW	-100 dBm	7000 m
XBee-ZigBee	Zigbee-Pro	2.4GHz	2 mW	-96 dBm	500 m
XBee-ZigBee-Pro	Zigbee-Pro	2.4GHz	50 mW	-102 dBm	7000 m
XBee-868	RF	868MHz	315 mW	-112 dBm	40 km
XBee-900	RF	900MHz	50 mW	-100 dBm	10 km
XBee-XSC	RF	900MHz	100 mW	-106 dBm	24 km

A web interface, as illustrated in Figure 3.5, was developed to monitor, configure and access the sensor data collected. Since each mote was also equipped with a GPS module, the position of every sensor node and possible detected events is known to the system manager or overseer. The collected data may be accessed in a more visual and user friendly way, which considerately increases the understanding of the covered area status.

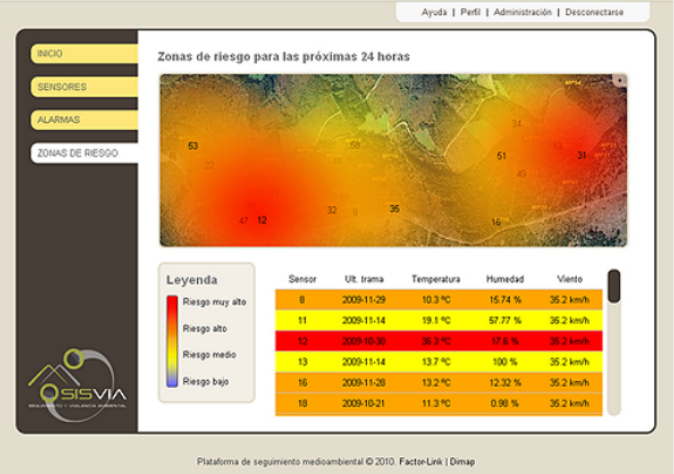


Figure 3.5: SISVIA Vigilancia y Seguimiento Ambiental Web interface.

3.2 FFSS - Forest-Fires Surveillance System

The FFSS - Forest-Fires Surveillance System was developed in 2006 with the purpose of monitoring the South Korean mountains for wildfire events in real-time. The purpose and main elements of this system are very similar to the one under study in this dissertation. The FFSS system is composed by the deployed WSNs, the middle-ware program and the web application, [26].

This project has the particularity of deploying several WSNs on different locations and recover data from all of them. Each WSN was developed in the same way, using the same components.

3.2.1 FFSS Sensor Node

As the main element in a WSN, the sensor node allows for collecting and sharing data from the surrounding environment. For fire detection purpose, there is a sensor combination of temperature, smoke, acoustic, olfactory and acceleration sensors.

At the core of the sensor node, FFSS uses TIP50CM, an ultra-low power wireless module for sensor networks, monitoring and property applications. It transmits on the 2.4GHz band and allows for a Single Frequency Network (SFN) creation. The specifications of the Sensor Node, Figure 3.6, can be found on Table 3.2.

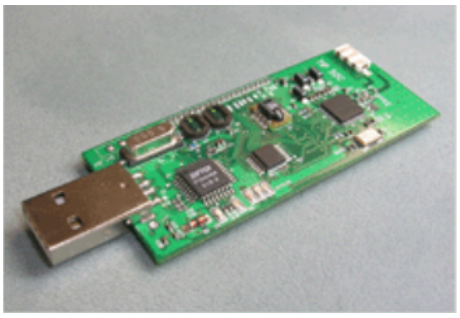


Figure 3.6: FFSS Sensor Node.

Table 3.2: FFSS Sensor Node Characteristics

Item	Description
Processor	16bit RISC, 8MHz256KB
Memory	256KB Program Flash
Operating System	Tiny OS
Multi-channel Radio	2,4GHz
Data Rate	250Kbyte
Sensor	Temperature, Humidity and Light
Network	Multi-hop and Ad-hoc
Interface	USB (UART)
Size	68x29mm
Power	3.0∼3.3 V
Range	70m in lab

3.2.2 FFSS Sensor Node Operating System

The FFSS sensor nodes have the applications running on a light, event based OS that is called TinyOS that upon receiving some sort of stimuli executes a determined function. The applications for this OS are developed in the NesC language, a language with great similarities with C programming language.

3.2.3 FFSS Network Protocol

This system uses a network protocol similar to MCF, Minimum Cost path Forwarding, routing protocol. This protocol finds the shortest path from all sensor nodes to the base station. This protocol is probably not the most adequate for this type of network since routing all the data from each node may create bottleneck or congestion issues on the upstream nodes resulting in the loss or corruption of data from monitored areas. In order to overcome this issue a limit on the expended energy for each round in each node was set. The main goals for the routing protocol were: Optimality, Simplicity and Scalability. Figure 3.7 describes how this protocol operates.

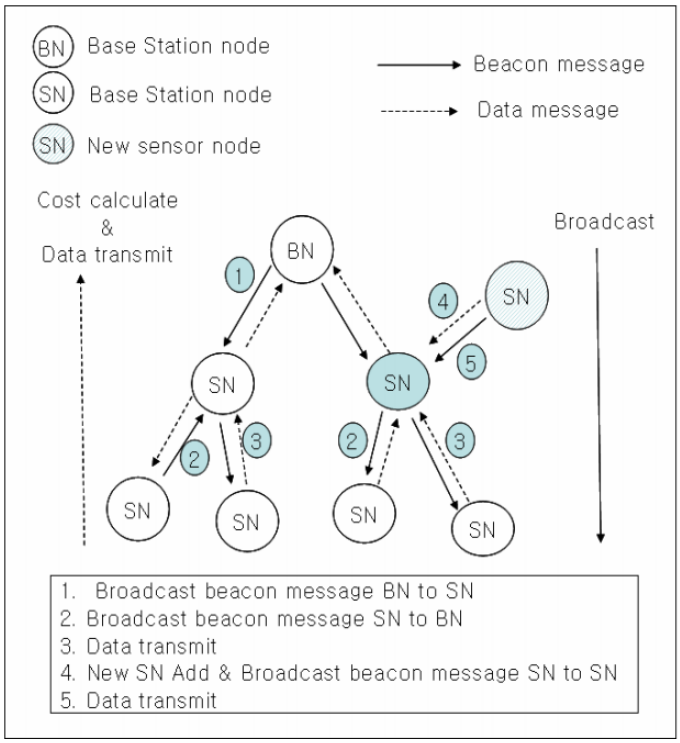


Figure 3.7: FFSS Configuration Step of the FSSS Network Protocol.

3.3 "Monitorização Ambiental em Espaços Florestais com Rede de Sensores Sem Fios" - Tiago Braga -2010

The project under review in this section is a system developed by Tiago Braga, for his master thesis, submitted in 2010 to *Universidade da Madeira Centro de Competência de Ciências Exactas e da Engenharia*, [10]. The proposed and developed system objectives were to develop and deploy a monitoring system by using wireless sensor network technology.

The developed system is based on the Zigbee Protocol and each deployed sensor node collects and sends data throughout the Zigbee network to a base station, where the data is processed, handled and stored in a database. The main variables and data collected during this project were:

. Temperature

. Humidity

. Luminosity

. RSSI - Received Signal Strength Indicator

A prototype was developed and deployed in an outside environment. The full system architecture applied to this project is illustrated in Figure 3.8.

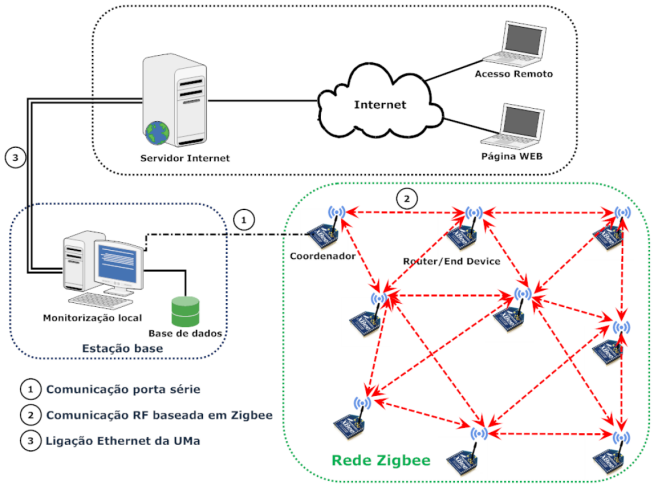


Figure 3.8: System Architecture [10].

For this project, the author has created his own sensor node. For this purpose, the same requirements mentioned in Section 2.2.2 were considered and a mote was created with an Atmega168 Micro-controller and a Xbee communication module at its core. Together with those basic components, to create a fully working sensor node, the full list of hardware or components was used is as follows [10]:

. **Battery** - The main source of energy are a pair of rechargeable AA batteries.

. **Step-up DC-DC Converter MAX1675** - Its main purpose is to supply a constant output voltage.

. **Solar Panel - MSX- 005F** - Used as means of producing energy to recharge the batteries

. **On-Off Circuit** - The main circuit used to regulate the battery charging.

. **Step-Down DC-DC Converter TL2575** - Converts an input voltage between 4.75V and 40V to 3.3, 5, 12 and 15V on the output depending on the resistances applied to the voltage divider.

. **XBee Radio Module** - The main RF module used to communicate between nodes. Zigbee compliant.

. **ATMega168 Microcontroller** - The sensor node main processing unit.

. **SHT15 Sensor** - Temperature/Humidity Sensor

. **S1087 Sensor** - Photodiode responsible for collecting luminosity data

Figure 3.9 describes the developed sensor node architecture and components.

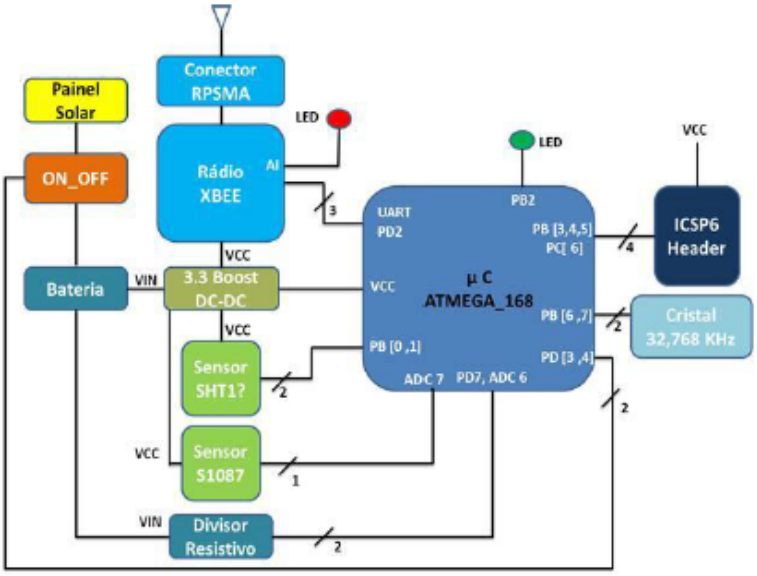


Figure 3.9: Sensor Node Architecture [10].

In terms of the network deployment, the prototype system was arranged in a straight line fashion where each node can only connect to the next in line, the only one in range. A visual description of the full deployed system can be found in Figure 3.10.

Figure 3.10: Geographic Distribution of Sensor Nodes [10].

In order to provide better autonomy to the network elements, sleep cycles were applied to the sensor network elements allowing for energy saving and network life extending, according to [10]. For that purpose, the sensor node deployed had the capability of switching from an end-device to a router working mode. Only end-devices are allowed to sleep, but end-devices can not forward data.

Chapter 4

Project Development

Considering the mentioned works in section 3, there was a need to develop a system by changing working procedures, applying more recent technologies and creating a more purpose oriented and reliable system with lower costs when compared to proprietary solutions. For that purpose, the main differences between the mentioned past works and the proposed system in this dissertation will be stated in this section along with some propositions for the development of this project.

When considering commercial products and solutions such as the project developed by Libelium together with DIMAP, the requirement for using their proprietary hardware and systems comes with high costs, in the range of tens of thousands of Euros. One of the main improvements proposed by the project under study is to reduce the costs of the development and deployment of a similar system without compromising its main functionalities and purpose oriented requirements. By acquiring or developing open-source devices with built-in sensors and communication modules, such as Telosb, Micaz, or similar, the sensor node costs drop significantly by reducing the need to spend an extra amount of funding in sensor boards or in developing the sensor node itself like the example described in Section 3.3. These devices also allow for a more compact solution, diminishing the deployed system's footprint on the monitored areas.

For the project at hands, the greatest improvement from the system proposed in [10] is an increase in the network reliability and fault tolerance. Instead of setting the sensor nodes in a straight line, the monitored area will adopt a grid topology where several nodes are in communication range of each other. This allows for greater fault tolerance since in the proposed system all nodes will have more then one way to acquire a path to the sink node. In the mentioned past work, by arranging the system in a straight line, in case of failure in one of the sensor nodes, the nodes further away will lose the connection to the network and sink node.

Another core feature that is distinct between this project and the mentioned ones is the application of the RPL protocol, specially designed for this type of network, LLN, that is expected to introduce some improvements in the network routing.

In terms of creating a system that is more oriented to fire detection, this project will feature the application of an alert state on a sensor node when its collected data is considered to reveal a possible danger situation. When the alert state of a node is set, its reading rate will be increased

according to the level of a calculated danger index. The system will report periodically in regular conditions, but when an alert state is set, considering its level, the sensor node may be set to collect data in real time in order to obtain extra data during danger situations.

This project objective is then to create a system consisting of a wireless sensor network, based on MTM-CM5000 devices, and a Web server and application. The WSN will monitor a determined area and collect data from the environment such as temperature, humidity and luminosity. The collected data, upon arriving on the sink node, will then be sent to a gateway that by running a script will receive and process the data, store it in a local file, send it to a web server, calculate and send to the sink node a danger index that allows for introducing a variable rate on a sensor node with high risk levels for a fire occurrence. The web server's purpose is to handle all client requests and all database operations. The web application will be responsible for presenting all the collected data upon request by a web client or browser.

With the above project specifications in mind, the development of this project was divided in two main components, the WSN and the Web application development. In the next two sections, the development stages and work will be described.

Considering the first challenge, choosing and acquiring IEEE 802.15.4 compliant devices, INESC Porto provided ten MTM-CM5000 motes that would be crucial for an experimental prototype of the system in mind. In order to develop the sensor network, a deep study and familiarization with the motes operating system was required. The MTM-CM5000 motes, as TelosB / Tmote Sky based motes, are compatible with both TinyOs and Contiki Os. In the beginning, a familiarization approach to TinyOs was done, but later, Contiki Os was chosen for the motes' main operating system due to support and integration with Cooja, a WSN simulator/emulator that had a great impact during the development stages. Contiki Os and Cooja will be analysed in greater detail in Section 4.2.1.

For the Web application deployment, free test services were used, Heroku for the web server and application and MLab for the database. To achieve a fully working system a bridge between these two components was also created by developing a Python script that connects the serial port of the sink node to the web server for storing, presenting and processing data, allowing data communication between these two different environments.

The following sections will give a detailed description of this project's development.

4.1 Project Network Analysis

In this section a data payload analysis will be made regarding the amount of information to be transmitted by each node upon the data collection by each sensor node. The decisions made in this section will have Figure 2.6, in Section 2.3.3, as its core, together with all the required information a sensor node will periodically collect and transmit.

Upon initialization of the network, each node will communicate the readings from each sensor and other required information.

The amount of information to be sent may have an impact on the network life time and performance, therefore all the nodes will transmit as little information as possible while still allowing the implementation of all the required features for this system.

Since the three built-in sensors on the CM5000 device are used and that includes temperature, humidity and luminosity, 2 bytes per sensor reading are required [27] [28] . In order to test results and keep track of the number of packets sent by each client or sensor node, another 2 bytes are required. To find what routes the packets are going through and allowing to draw a dynamic network diagram on the web application, we must also require the parent ID of each node, resulting in one extra byte. Another feature will be to allow the motes to enter an alert state whenever a possible emergency situation is detected. For that purpose, a one byte variable containing information about the current rate on a determined node is also required. After all the above considerations, the data payload for each packet, without any separation, is as shown in Table 4.1.

Table 4.1: Payload Required Data

seq_id	Temperature	Humidity	Luminosity	Parent_ID	Rate
2 bytes	2 bytes	2 bytes	2 bytes	1 byte	1 byte

When compared with the general data frame for 802.15.4 protocol, it can be considered that the information each node will transmit is a small fraction of the maximum size a data frame may take. With a maximum MAC frame size of 127 bytes and a maximum header and footer size of 25 bytes, the data payload may be of at least 102 bytes [29],[11],[3].

Figure 4.1 describes the general data frame format for IEEE 802.15.4 standard and, together with Figure 2.6, describes in more detail all the MAC header fields.

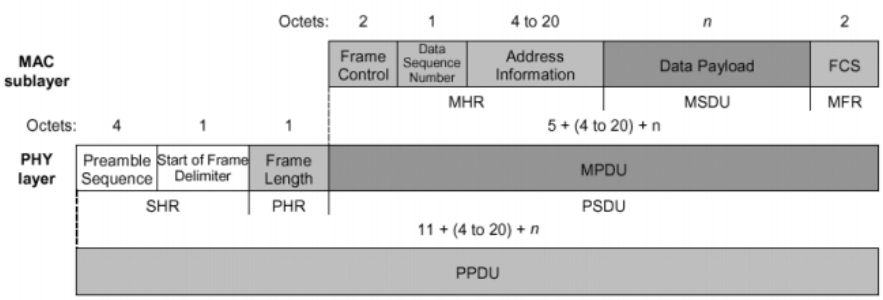


Figure 4.1: IEEE802.15.4 Data Frame [11]

By analysing the transmitted packets logged during a simulation of the developed network in Wireshark, as illustrated in Figure 4.2, it is possible to see that the collected data at the sensor node occupies 19 bytes, containing the above mentioned information with added separators between each variable. Considering that the collected data at a sensor node is converted to a string before being sent, the 19 bytes correspond to 1 byte per character including the separators.

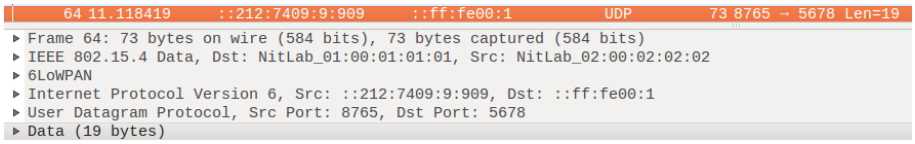


Figure 4.2: Wireshark capture of sent data frame.

From further analysing the sent packet, other informations can also be retrieved. For instance, the full size of the sent packet, including the mentioned data payload, is 73 bytes. This includes all the addressing, frame control, frame sequence and other fields shown in Figure 2.6.

4.2 WSN Development

The following sections will describe the developed sensor node network and its development fundamentals.

4.2.1 Contiki OS and Cooja

As mentioned before, initially some time was spent in familiarizing with Tiny OS but later the whole network development has shifted to Contiki OS. The main reason for this change was the support for the WSN simulator/emulator Cooja, that comes bundled with Contiki OS. There are other aspects to this change that will be further developed in the following sections.

Contiki OS is an "Open Source OS for the Internet Of Things" and provides support for both IPv4 and IPv6 and related low-power wireless standards such as 6lowPAN and RPL. RPL, as mentioned in Section 2.5.2, is a protocol specially designed for low power, low rate WSNs. Although Tiny OS already has such compatibilities, they are harder to implement.

When programming a mote in Contiki OS, the standard C language is used for the applications' development. The developed applications in Contiki may be uploaded to a real physical device, or, with no changes to the code, emulated on Cooja. Both of these ContikiOS characteristics introduce faster coding and testing of the developed applications.

Cooja is a simulator developed with Java that allows to simulate or emulate wireless sensor network deployments. It is capable of emulating several different devices whose code was developed in Contiki OS. When simulating a device or network in Cooja, the behaviour real devices would take is emulated. This allows the developer to keep compiling and testing code without the need of programming actual devices, which is a more time consuming approach. These are the main reasons that make Cooja a great development tool for these type of networks. It is important to also mention that the devices utilized, MTM-CM5000, as Telosb/Tmote Sky family mote, are fully compatible with Cooja.

Another strong point for this simulator is the ability for users to develop their own plugins to their own needs. For this project, two community developed plugins were used when simulating and testing the network. The external plugins or tools added and used in Cooja for developing this

project are Serial2Pty, that allows to connect a simulated node to a virtual serial-port, and Sky Gui that provides the possibility to alter simulated luminosity values with an interactive slide.

Serial2Pty was of great importance when developing the script responsible to act as a connection between the sink node and web server. Without it, upon testing the connection between these two parts of the system, every device code change would have to firstly be applied to the sink or client nodes as needed, greatly increasing the developing time. Sky GUI was mostly used when developing the variable rate change applied to the alert mode. Once again, this plugin saved a great amount of time when testing and developing this feature.

Besides these plugins, Cooja has some pre built tools and plugins that were also useful, Figure 4.3, shows the built-in and user developed tools and plugins that were used during the development and testing stages of this project.

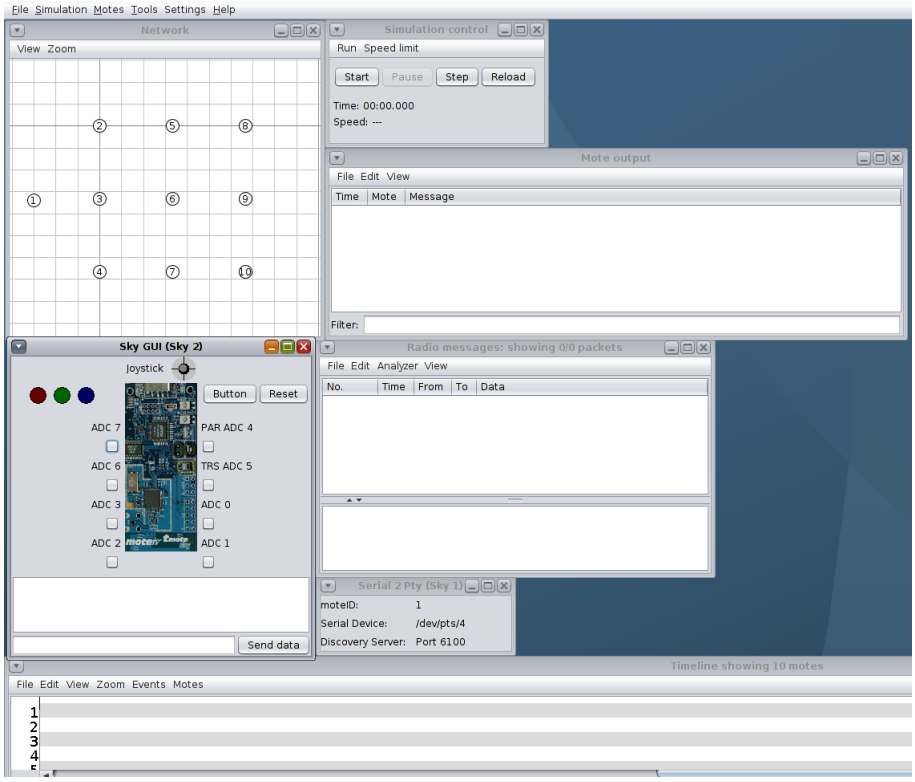


Figure 4.3: Sink Node Functional Diagram

. **Network** - Shows the devices position, allows to remove and move motes and control mote specific tools,

. **Simulation Control** - Allows to Start, Pause, Reload and control the simulation speed,

. **Mote output** - Shows the output of every mote in the network. Allows for filtering,

. **Radio Messages** - Lists all the transmitted radio packets during the network simulation,

. **Sky GUI** - Allows altering simulated sensor data,

. **Serial 2 Pty** - Creates a virtual serial port on the stated address,

. **Timeline** - Shows selected events on motes in a timeline, radio traffic, radio toggling on or off, mote LEDs and others.

Together with the integration with Cooja, Contiki also includes upon installation a wide range of examples for most of the supported functionalities that allow a faster learning of the OS and the specific methods necessary to develop the intended project.

In order to develop the proposed WSN, two main entities were required to be present: the sink node and the client nodes. The following subsections will explain the development methodologies and the final working functionalities for both these entities.

Starting with a ContikiOS provided example for creating a simplistic WSN using RPL and UDP connections, the IPv6 RPL-UDP example, both the sink and client nodes were developed with it as its core.

4.2.2 Sink Node

The sink node is one of the core elements in a network of this type. It is responsible for creating the whole network and receiving all the client collected data. This node, connected with a gateway through a serial port, or other, passes the collected data that is later sent to a web server to be processed, stored in a database and presented in the Web application.

As mentioned before, the sink node development is based on the Ipv6 RPL-UDP example provided in Contiki that allows for Hello messages to be sent from any client in the network to the sink node.

With a good starting point, the server had already some of the main required functionalities implemented such as the network creation, IPv6 attribution, network repair upon pressing the mote user button and the full routine to handle incoming packets and print to the serial port by using a regular printf routine.

Changes were introduced in order to allow the gateway device to communicate with the sink node. By including a method that upon receiving any byte on the serial port, sent by the gateway script for setting a new reading rate on a node, a routine is called to process the received data and forward it to the corresponding client node.

Other implemented feature is the network repair function based on a timer, to ensure that network routes and changes are automatically detected. The decision to implement this periodic auto repair feature was made after some tests with real devices. It was observed that upon moving a node and forcing it to change its parent by moving it in and out of range of other nodes, the client preferred parents information wasn't immediately updated but upon running the network repair the network information was all up to date. This periodic repair makes sure that no old network or routing information comes from the client nodes.

In terms of network and network routing protocol, all the options available in Contiki and pre implemented on the example code remained unchanged. The network formation transmitted

packets between the sink node and a single client node can be observed in the following Wireshark log, allowing for a better perception of how a connection between sink and client nodes is accomplished:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::212:7402:2:202	ff02::1a	ICMPv6	64	RPL Control (DODAG Information Solicitation)
2	2.991663	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
3	5.125911	fe80::212:7402:2:202	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
4	7.712050	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)
5	7.714898			IEEE 802.15.4	5	Ack
6	10.131660	fe80::212:7401:1:101	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
7	12.032733	fe80::212:7402:2:202	ff02::1a	ICMPv6	97	RPL Control (DODAG Information Object)
8	13.396278	::212:7402:2:202	::ff:fe00:1	UDP	64	8765 → 5678 Len=19
9	13.398744			IEEE 802.15.4	5	Ack
10	14.305776	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	76	RPL Control (Destination Advertisement Object)
11	14.308626			IEEE 802.15.4	5	Ack
12	16.002927	fe80::212:7402:2:202	fe80::212:7401:1:101	ICMPv6	102	RPL Control (DODAG Information Object)
13	16.006608			IEEE 802.15.4	5	Ack
14	23.903870	::212:7402:2:202	::ff:fe00:1	UDP	63	8765 → 5678 Len=18

Figure 4.4: Network creation transmitted packets between sink and client node

To conclude the sink node development analysis, Figure 4.5 block diagram represents the working flow implemented to the sink node.

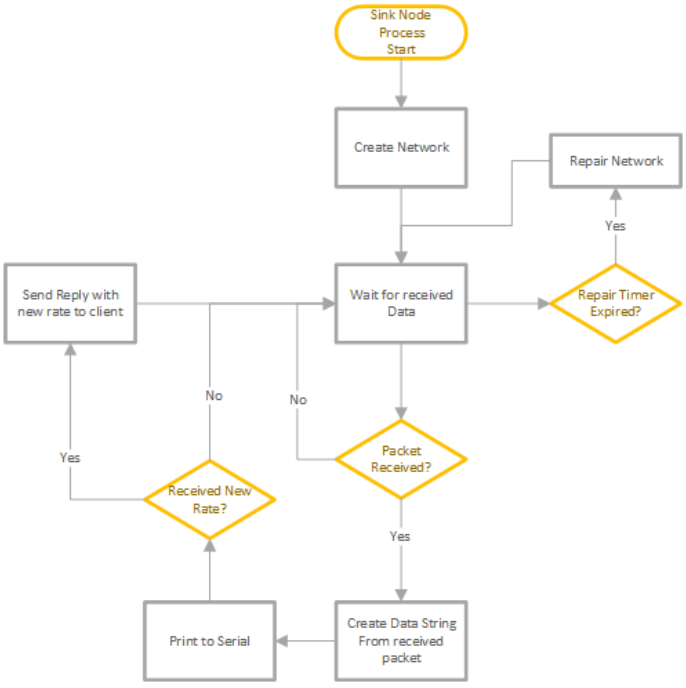


Figure 4.5: Sink Node Functional Diagram

4.2.3 Client Node

Client nodes are just as important to the proposed system as the sink node as without them no information is sent to the sink node. These nodes are responsible for obtaining and sharing the desired environment data in a monitoring system. By using the built-in sensors in the MTM-CM5000 motes, the client nodes' basic function is to collect the air temperature and humidity values, as well as luminosity. For all the intended features for the whole system, other information was also required to be sent from the client node, as described in Section 4.1, but, initially, the true core function for these nodes was to collect and send the sensor readings to the sink node.

As with the sink node, the development of the client code was started with the "IPv6 PRL-UDP" Contiki example as its core. The code was gradually altered and expanded to comply with all the intended requirements. The first change to be applied was made to complete the main objective for these devices, implement a method that allows for collecting and sending sensor data to the sink node.

By keeping all the default network properties, the only steps required to achieve basic functionality were to activate the sensors, set periodic timers to control the rate at which sensors are activated and sensor data is collected. When the data acquisition is finished, the client node constructs a message containing all the information required and sends it to the sink node.

To allow for the reception of a new rate from the sink, an event handler was implemented to be ran upon receiving a packet from the sink node.

To be able to implement some other features, the client node was also required to add the ID of its current preferred parent. This value was later used both in the web application and system testing and results, for it allows to know the path sent packets take in each transmission. To obtain this value a function obtains the preferred parent from the client node DAG structure, which contains routing information pertinent to the node.

Figure 4.6 represents the code logic applied to the client devices in a block diagram.

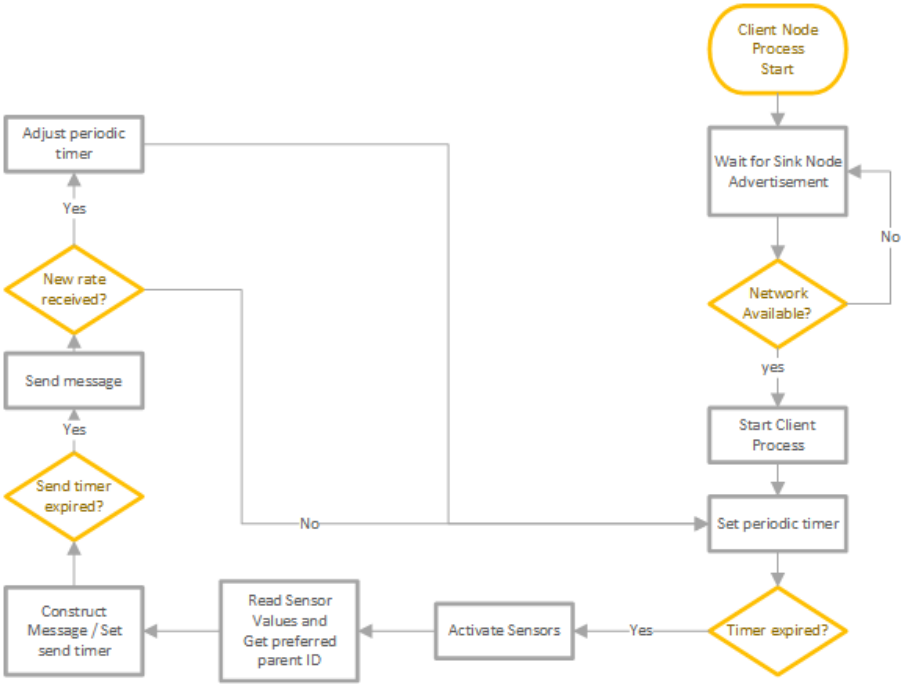


Figure 4.6: Client Node Functional Diagram

4.2.4 Serial to Web Communication

With the whole network collecting and sending data to the sink node, it was necessary to find the means for sending that data to the web server. With that purpose in mind, a Python script was

developed to allow the communication between the physical WSN and the Web application.

Upon starting, the script connects to the sink node serial port and waits for any output from the mote. As soon as new data is detected in the serial connection, it is collected, processed and through an HTTP POST request, sent to the web server. The danger index of the network mote is calculated and, if necessary, an alteration to the specific client node reading rate is issued to the sink node, which then forwards that command to the client in question. In this project an Ubuntu PC was used, but considering the project's aim, Python was used with a more compact size gateway in mind, as it is widely used across several popular platforms and is compatible with a lot of so called micro-computers such as the Raspberry Pi.

The diagram illustrated in Figure 4.7 exemplifies the operating sequence that the script is designed to execute.

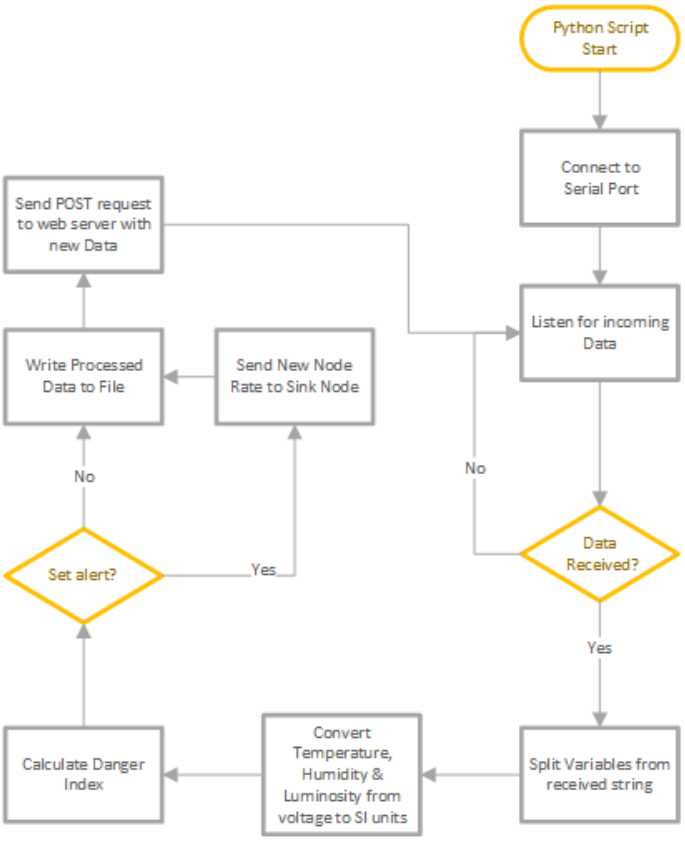


Figure 4.7: Python Script Functional Diagram

4.2.4.1 Data Handling

Upon receiving the printed message from the sink node in the serial port, some operations and conversions must be applied since all that is collected is a single string. To convert temperature, humidity and luminosity values from voltage to SI units, the following formulas, found in the sensors datasheet, were applied to each collected value [27], [28].

What makes it necessary to execute these calculations in this script instead of locally in each sensor node are the memory constrains on the devices and the intent to reduce each sensor node working load to its minimum. The following tables express the formulas used to convert each physical value to SI units.

Table 4.2: Linear Relative Humidity conversion formula

Linear Relative Humidity (%)
$RH_{linear} = -4 + 0.0405 * SO_{RH} - 2.8e^{-6} * SO_{RH}^2$

Table 4.3: Air Temperature conversion formula

Air Temperature (ºC)
$T = -39.6 + 0.01 * SO_T$

For temperatures to far apart from 25 ºC, the humidity signal requires a compensation. Considering the nature of the project, to detect fire occurrences, the air temperature may well be far from that mark, for that reason the following correction formula for the linear relative humidity was also applied.

Table 4.4: Relative humidity compensation for temperature formula

Relative Humidity - compensated (%)
$RH_{true} = (T_{(^oC)} - 25) * (0.01 + 0.00008 * RH_{linear})$

Table 4.5: Luminosity conversion formula

Luminosity (lux)
$L_{lux} = 10 * \frac{L}{7}$

4.2.4.2 Danger Index Calculation

With the required feature for changing a node behaviour upon a possible danger event in mind, a danger index calculation must be applied to determine when to set an alert state to a specific mote. The type of information the sensor nodes are able to collect and transmit limit the choices for applying a fire danger index with high complexity and accuracy.

Considering that the MTM-CM5000 motes can only collect temperature, humidity and light values, the chosen fire danger index was the Angstrom Fire Index, a Swedish index [30]. This index only bases itself on information about the air temperature and humidity to calculate the likelihood of a fire occurence. The Angstrom fire index calculation is as follows [30]:

$$I = \frac{R}{20} + \left[\frac{(27 - T)}{10}\right] \tag{4.1}$$

where:

. **I**= Angstrom Index

. **R**= Relative Humidity (%)

. **T**= Air Temperature (^{o}C)

Interpretation:

. **>4** : Fire occurrence unlikely

. **4.0-2.5** : Fire conditions unfavourable

. **2.5 - 2.0** : Fire conditions favourable

. **<2.0** : Fire occurrence very likely

By analysing the formula it is possible to determine that the relative humidity is the factor that most impacts this index and the temperature impact is almost insignificant for certain values, with only temperatures well above 27^{o}C increasing the likelihood of a fire occurrence slightly. This calculation, for the designed system, is done in the Python script responsible for processing the received data at the sink node, as mentioned above in Section 4.2.4.

4.3 Web Application Development

To develop a web application that complements the WSN and allows for accessing and presenting the collected data, a web server and client were created. The web server together with the web services were developed with Node.js and Express frameworks at their core. The web client was created using a JavaScript framework, Angular JS and some extra plugins and directives were also utilized. For this development, a previously developed Web API, with similarities to the one desired for this project, was found described in a master thesis project, published for *Mestrado em Engenharia Informática - Universidade do Minho*, submitted by Marcos Magalhães [31]. In a collaboration effort between him and the author of this project, a new web infrastructure was developed with similar server and application logic, tools and frameworks. This section aims to describe the development methodology applied to the entire web application, to provide more information about the used tools in this project and lastly to demonstrate the final look and process flow for the whole application.

4.3.1 Web Server and Service

In order to create a full web application for the proposed system with all the required features, as it was specified in Section 4, there was a need to develop a web server capable of processing and handling the data collected from the developed WSN for further presentation in the web application.

The web server handles HTTP POST and GET requests from the client who awaits the server reply. The server is also responsible for database control and access if a client request requires it.

The server was implemented to support a RESTful architecture that amongst other properties is able to prevent conflicts at the server when access is requested by multiple clients. As mentioned before, the chosen framework for the web server was Node.js since its operation is event oriented, preventing it from malfunctioning during input and output operations.

The implemented web services for the system were based on the framework Node.js that, together with the framework Express, allowed for the development of a system that is able to handle requests, read and write from the database. To create the database for storing the desired information and allow the web server access, mLab services were used and a database was set up online.

To include a login functionality and handle all the authentication process the Passport framework was used.

After authentication of a client, the framework Socket.IO enables a tunnel creation between server and client, allowing for data exchange in real time.

The web server for this project, is responsible for handling the incoming POST requests sent by the script connected to the sink node from the WSN and writting them to a database.

4.3.2 Web Client

The web client, basically a JavaScript program running on a web browser, was developed by using the JavaScript framework AngularJS. This framework was developed with the intention of reducing complexity and server bandwidth usage. Also, AngularJS allows for a better differentiation between implemented browser functionalities. Data transmitted between the web client and application is made through the usage of JSON packets, which simplifies data exchange to a great extent.

The web client's main purpose is to present the collected data from the WSN that was sent to the web server and stored in the online database. For this purpose, the intended web client was developed to present three main elements.

The first element is a table containing all the records stored in the database, each entry in the table corresponds to an individual packet sent by a sensor node as illustrated in Figure 4.8.

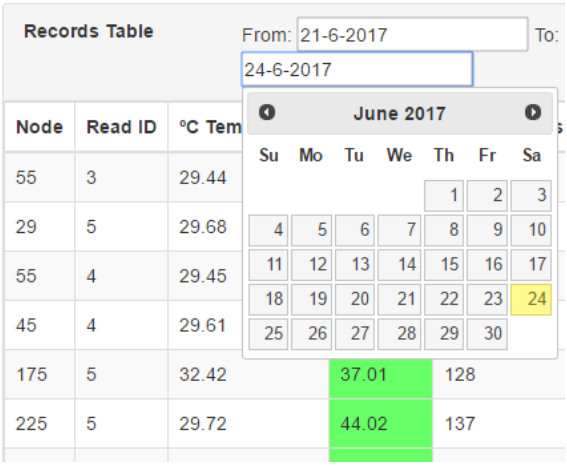


Figure 4.8: Table for presenting received entries and data picker to apply date filter.

The second development was to translate that data to charts and, using Socket.IO functionalities, to update the charts' data in real time, when the whole system is in execution. The charts' functionalities were implemented with Highcharts-ng, an AngularJS directive for Highcharts. Figure 4.9 shows the Luminosity chart. After the integration of Socket.io with Hicharts-ng directive, the charts are updated in real time, as long as there is already one entry for each node IDs.

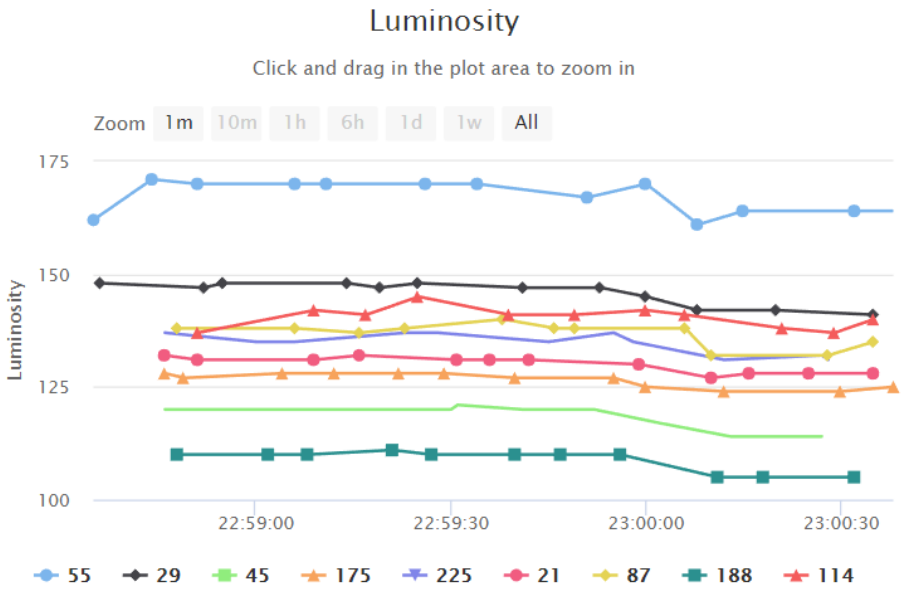


Figure 4.9: Highcharts-ng directive for rendering Luminosity values in real-time.

The last feature implemented for the web application and displayed web page provides the ability to draw the network topology in real time by using VisJS angular directives. From the node ID and Parent ID information received from the each of the WSN's sensor nodes, the application is able to draw the inter connections between every node present in the network, displaying the actual network tree and connections, as illustrated in Figure 4.10.

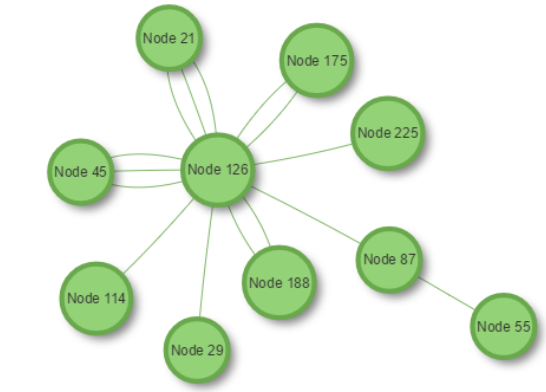


Figure 4.10: VisJS plugin rendering a Force-Directed Graph of the network.

To add an alert functionality, a filter was applied to the received data to reflect the WSN alert state. The same threshold that when crossed makes a client node go into alert mode was applied to the web application. Upon receiving a reading over the defined limit, a warning is shown in the web browser displayed page.

Both the created web application as well as all the implemented data presentation features allow for an easier monitoring of the network status and collected data, and alert generation for higher likelihood of a forest fire, as intended.

To conclude this section, Figure 4.11, a screen shot taken during the rate variation test, shows how the web client looks as the whole system is up and running and an alert is being issued.

Figure 4.11: Web client interface during rate variation simulation.

Chapter 5

Results

In this section the overall results of the project will be presented and discussed in each subsection. The tests will be applied to both simulation and real devices in order to show the correct operation in both virtual and real environments. Amongst other statistics the main purpose is to check the received packet ratio, the percentage of packets each node sent, to detect alterations in received packet number upon setting the alert state on one or more sensor nodes or on simulating a sensor node fault.

These tests' aim is to determine the correct behaviour of the network in normal operation mode, the impact of the alert mode feature on the total number of packets, the fault tolerance to a dropped sensor node and the expected behaviour of the whole system in these situations.

For all tests where alert mode and rate variation are focused, both in simulation and in real MTM-CM5000 motes, the Angstrom danger index calculation defined in Section 4.2.4.2 will not be used due to the difficulty of accurately reproducing the desired alterations in read values for real and simulation devices. Instead, controlled changes in the collected luminosity value will be applied.

By using Sky mote GUI plugin for COOJA there is the possibility to alter this value with an interactive slider, and for real devices adding a source of light or covering them is also enough to handle these tests. Altering temperature and humidity values would be especially complicated when using real devices compared to the proposed solution.

5.1 Simulation Tests

In this section the applied test methodology to all the simulations will be described. In order to obtain coherent results, a baseline test will be firstly done with all the WSN devices working properly and with no alert states. For the baseline and all subsequent tests, the simulations will have a runtime of 30 minutes, will have the simulation based on the same random seed in Cooja, will keep the same network grid topology, and the whole system will be in execution: all the sensor nodes, the sink node, the Python script, the web server and web application.

To make sure these parameters stay the same, a simulation was previously created and saved to allow for a new simulation to be loaded with exactly the same conditions. Figure 5.1 shows the Cooja GUI upon the test simulation loading. Each interface to be seen in this picture is already described in Section 4.2.1.

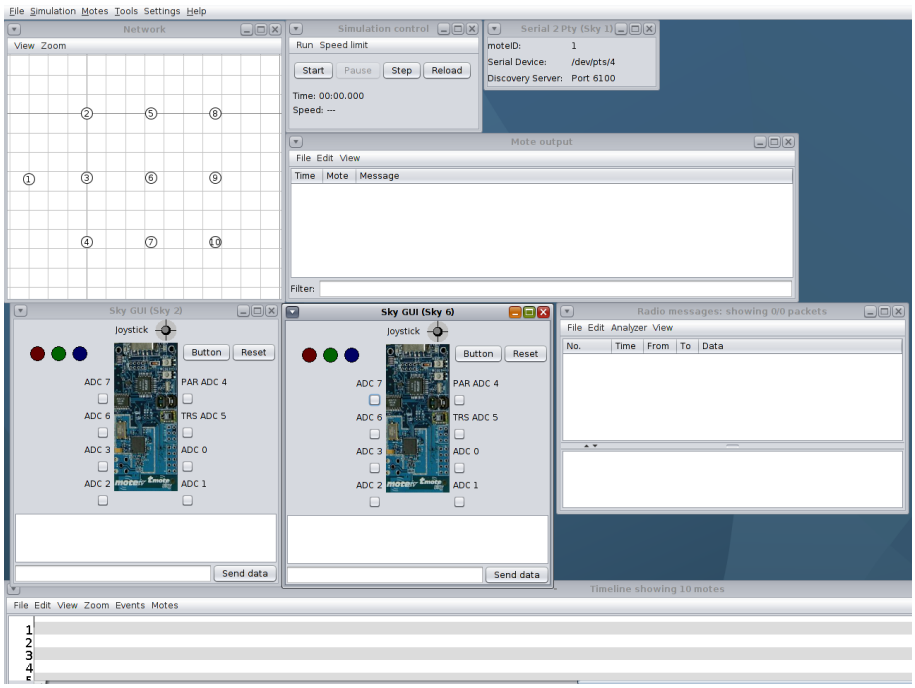


Figure 5.1: Cooja Baseline simulation interface

5.1.1 Simulation - Baseline Test

As mentioned before, this is the baseline test to which all the others will be compared against in order to obtain results and conclusions about the system overall performance.

This baseline test consists of running the previously set and saved simulation without changing any parameters during the whole test and verify the total number of received packets, both on the sink node and the web server. When compared to the expected number of received packets, this resulting data will allow to check if the system is able to collect and transmit all the information expected from client to sink node and from the sink node to the web database. As mentioned in Section 4.2.4, the developed gateway script is also responsible for recording all the collected data on a local log file. This file is a *JSON* file that can be converted to *Xls* file, a format compatible with Microsoft Office Excel that will allow for further analysis of the data. Considering the reading rate, the number of motes and the test duration, the following calculations will reveal the expected number of received packets.

. **From 0 to 30 minutes**

- 9 motes send 1 packet every 10 seconds.
- in 1 minute: $[9 * 6 = 54]$ packets are expected.

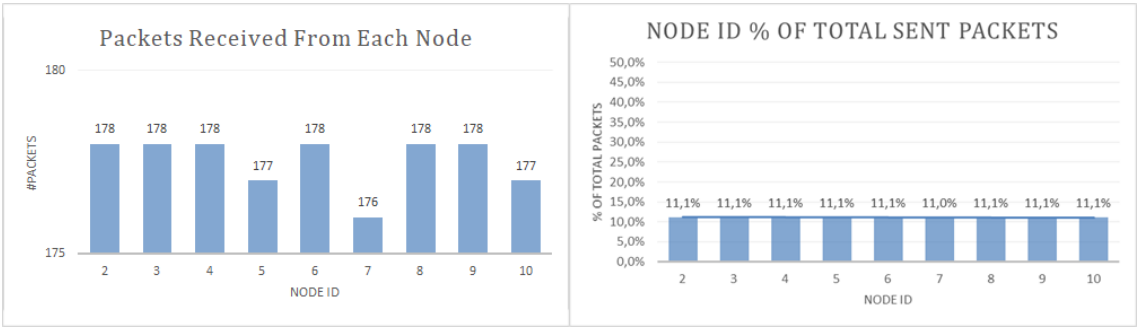
- in 30 minutes: $[54 * 30 = 1620]$ packets are expected.

Upon completion of the test, results were obtained for the baseline test that are detailed in Table 5.1:

Table 5.1: Number of Expected and Received packets

# Expected Received Packets	**# Packets Received**	**# Packets Received in Database**	**Ratio Received/Expected**
1620	1598	1598	98,64%

From the obtained numeric results after treating the collected information it can be observed that there are some differences between the expected results and the simulation ones. From a 1602 total received packets estimate, instead of the expected 100% ratio, only 99.75% of the expected packets were received. The explanation for this deviation is that upon the network start, not all nodes are immediately connect to the sink node, nor do they all start collecting data exactly at the same time. This fact, allied with the fact that the simulation finishes and does not wait for all the sensor nodes to send their last packet, some sensor nodes missed sending some packets. We can conclude that these results, despite the small deviation, demonstrate the correct operation of the network in a simulation environment. The charts represented in Figure 5.2a and 5.2b were obtained from the collected data and reflect the impact each sensor node has on the total number of packets transmitted during execution. The data represented in Figure 5.3 was obtained from the database management centre, and shows that all the collected packets were also forwarded and stored in the database correctly.



(a) Total received packets / Node Id (b) Percentage of total received packets / Node Id

Figure 5.2: Obtained charts from simulated data

NAME	DOCUMENTS	CAPPED?	SIZE
nodes	1,598	false	438.41 KB

Figure 5.3: Number of stored entries in the Database

For further testing in this topic, the frequency each node is selected as parent by another node is also an important data extrapolation that should be made in order to better understand what

nodes are used more often as relays for sent packets. With these results it is possible to determine which nodes have higher impact in the network when it comes to routing and forwarding packets. For this baseline test, with the above mentioned procedures, the chart in Figure 5.4 reflects the number of times each node was selected as parent throughout the whole simulation. The number of times the sink node was elected as parent was omitted in this analysis since it will always be selected as a parent on every instance of the execution, otherwise the network would fail.

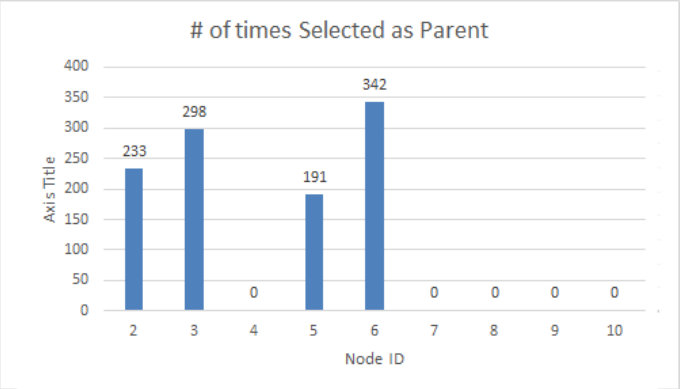


Figure 5.4: Number of times each node was selected as a parent node

In a regular working mode, throughout the whole simulation, only 4 nodes were actually selected as parents, especially node 6. This difference in the frequency at which this node is selected as parent can be explained by its position relative to the other network devices, by being in the centre of the network, it is responsible for forwarding a great amount of the total packets transmitted, out of the 1598 received packets, 542 or 33.9% where forwarded by node 6, originated from its direct children nodes alone. This result is important because in a real world deployment this would mean that node 6 has the highest load of processing and transmitting rate meaning its power consumption is also greater, its life time is shorter and therefore there is a higher probability of failure in this device.

5.1.2 Simulation - Variable Rate Test

For this test, the aim is to show that the application of alert states to one or several nodes will impact the rate at which they send packets and therefore the total number of received packets and the sent packet percentage of the specific node should also reflect this change.

For this test the following approach was used:

1. Start previously recorded simulation.
2. Run for a total of 30 minutes.
3. First 10 minutes no change applied.
4. From 10 to 20 minutes, set a sensor node luminosity value above defined threshold using Sky Mote tools GUI in Cooja.

5. From 20 min to the end, set another sensor node luminosity value above defined threshold.

Considering a total run time of 30 minutes and the sequence of steps mentioned above, the expected results in terms of received packets numbers are calculated in Table 5.2.

Table 5.2: Expected received packets calculations for variable rate simulation test.

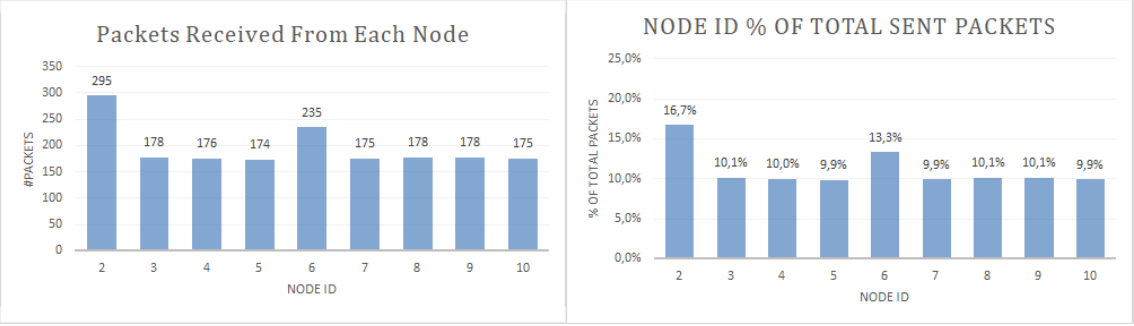
Time Period (minutes)	**Action**	**Expected Received Packet Calculations**
0 - 10	-	- 9 motes send 1 packet every 10 seconds - in 1 minute: $[9*6=54]$ packets are expected - in 10 minutes: $[54*10=540]$ packets are expected.
10 - 20	Raise luminosity on 1st mote	- 8 motes send 1 packet every 10 seconds - 1 mote sends 1 packet every 5 seconds - in 1 minute: $[8*6+1*12=60]$ packets are expected - in 10 minutes: $[10*60=600]$ packets are expected
20 - 30	Raise luminosity on 2nd mote	- 7 motes send 1 packet every 10 seconds - 2 motes send 1 packet every 5 seconds - in 1 minute: $[7*6+2*12=66]$ packets are expected - in 10 minutes: $[10*66=660]$ packets are expected

. **Total number of packets expected -** $[540+600+660=1800]$ packets are expected after 30 minutes.

Table 5.3, represents the obtained results after processing the data obtained from the full simulation. Figure 5.5a represents the total sent packets by each individual node and Figure 5.5b represents the percentage from the total received packets that each node sent.

Table 5.3: Number of Expected and Received packets

# Expected Received Packets	**# Packets Received**	**# Packets Received in Database**	**Ratio Received/Expected**
1800	1764	1764	98%



(a) Total received packets / Node Id (b) Percentage of total received packets / Node Id

Figure 5.5: Obtained charts from simulation data

As can be observed in Figure 5.6, all the received packets at the gateway were also stored in the database.

Figure 5.6: Number of stored entries in the Database

The minimal deviation between the expected and the obtained results is both for the same reasons stated in the baseline test, as described in Section 5.1.1 and due to a one cycle delay on each node upon changing the rate to an alert state. When the luminosity value is detected to be above the define threshold, the script sends the new rate to the sink node that then forwards it to the client node. When the new rate arrives at the client node, the periodic timer that regulates the reading rate has already been set again for the previous rate (10s) and only at the next cycle will it acquire the new period (5s).

Despite the mentioned factors, the described behaviour can be proved by analysing the above information. It is clearly visible that node 2 sent an higher percentage of the total number of packets and node 6 follows as the node with the next highest percentage. From that information alone it is possible to conclude that node 2 was set to be in alert mode from minute 10 to minute 30 and node 6 was set to be in alert in the last 10 minutes. These results prove that the assignment of a variable rate to an alert state node was implemented with success in a simulation environment.

5.1.3 Simulation - Fault tolerance Test

In this test, we address the network behaviour upon loosing one or more nodes, and its ability to recover from a missing node by changing the existent routes and convey the information through another path.

The applied methodology will be similar to the variable rate test, as described in Section 5.1.2 and will keep all the same initial conditions stated in Section 5.1.

Similarly to the rate variation test, this test will be ran for 30 minutes with the following procedures to be applied and expected received packets calculations shown in Table 5.4.

Table 5.4: Expected received packets calculations for fault tolerance simulation test.

Time Period (minutes)	**Action**	**Expected Received Packet Calculations**
0 - 10	-	- 9 motes send 1 packet every 10 seconds. - in 1 minute: $[9*6=54]$ packets are expected - in 10 minutes: $[54*10=540]$ packets are expected.
10 - 20	Remove 1 mote	- 8 motes send 1 packet every 10 seconds - in 1 minute: $[8*6=48]$ packets are expected - in 10 minutes: $[10*48=480]$ packets are expected
20 - 30	Remove 2nd mote	- 7 motes send 1 packet every 10 seconds - in 1 minute: $[7*6=42]$ packets are expected - in 10 minutes: $[10*42=420]$ packets are expected

. **Total number of packets expected -** $[540+480+420=1440]$ packets are expected after 30 minutes.

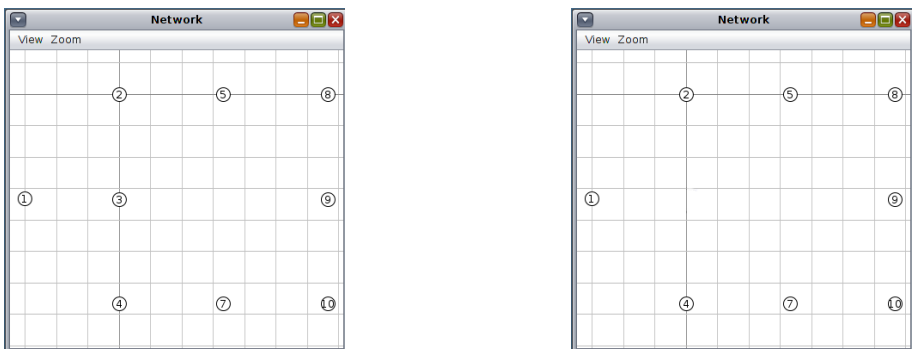
Table 5.5 shows the results for the expected and the received number of packets.

Table 5.5: Number of Expected and Received packets

# Expected Received Packets	# Packets Received	# Packets Received in Database	Ratio Received/Expected
1440	1405	1405	97.57%

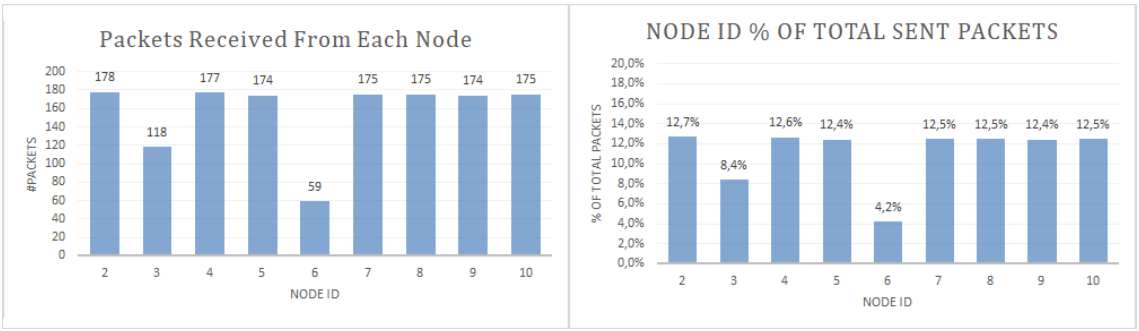
Once again, there is a small deviation between the expected to receive and actually received number of packets. This deviation is again a symptom related to the motes start time being different and to the fact that upon removing a mote and when stopping the simulation, due to the lack of synchronization, it is possible that a sensor node read cycle may be unfinished and therefore, no packet will be sent. Also, for this specific test, a lower ratio between the actual received packets and the number of expected received packets, calculated above, was to be anticipated. In above calculations for expected received data, the packets already sent, at the moment of the node removal, from nodes whose parents disappeared, were not accounted for. The previously mentioned unaccounted sent packets were dropped the moment or following moments after the parents were removed, resulting in a lower ratio between the expected and the obtained results, when compared to previous tests.

By analysing the results in Figure 5.8, generated from the collected data, it can be observed that node 6 and node 2 had a lower percentage of the total packets received at the sink. Considering that node 6 has an even lower number of sent packets than node 2 and, therefore, a lower percentage of the sent data, it can easily be observed which node was removed first, in this case, it is simple to extrapolate that node 6 was disconnected at the 10 minute mark and node 2 was disconnected only at the 20 minute mark. In fact, these were the exact steps applied and Figure 5.7a and 5.7b correspond to the network diagram during the 10 to 20 and 20-30 minute periods, respectively.



(a) Network Diagram - 10 to 20 minute period (b) Network Diagram - 20 to 30 minute period

Figure 5.7: Obtained charts from data



(a) Total received packets / Node Id (b) Percentage of total received packets / Node Id

Figure 5.8: Obtained charts from simulation data

As shown in Figure 5.9, all the packets received at the gateway were also successfully stored in the database.

NAME	DOCUMENTS	CAPPED?	SIZE
nodes	1,405	false	385.19 KB

Figure 5.9: Number of stored entries in the Database

From analysing the generated test data and subdividing it in three 10 minute periods, coinciding with the node removal moments, it was found that the number of times each node is selected as a parent was considerably altered as each node was removed. Figure 5.10 reflects the conclusions by showing how many times each node was chosen to be the parent of another node. The observed change in behaviour, when compared to the analogous results during the baseline test, proves that the network routes were dynamically altered throughout the test, as expected.

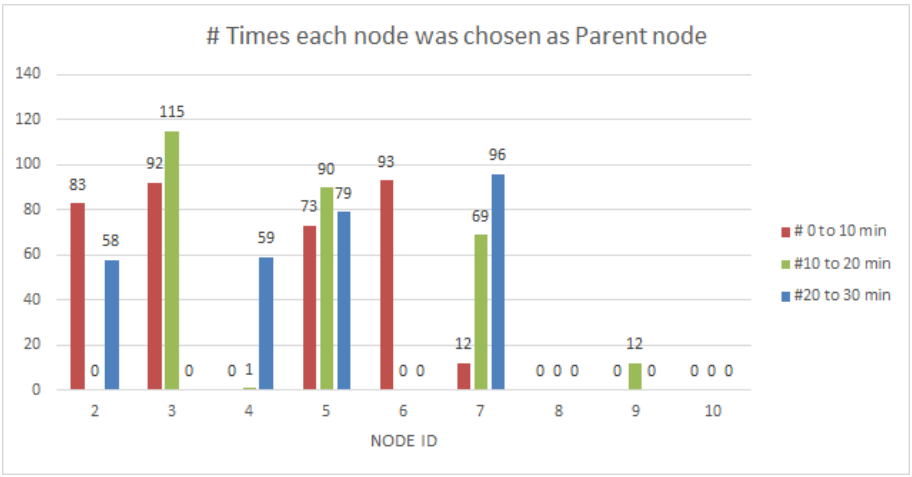


Figure 5.10: Number of Times each node was selected as Parent

When compared to the baseline test and its received/expected ratio of 98.64% during what is considered the regular operation mode, it can be extrapolated that the procedures applied in this test, removing two of the most used as parent motes, lead to an increase of 1.07% of packet

loss. Considering this is the worst or one of the worst scenarios, to have a loss of 1.07% of packets contributes to almost no loss of information meaning it can be considered that the system performance in recovering from a missing node was good.

In conclusion it was observed that the fault tolerance test, although still with an high ratio between expected and obtained results, was the test or situation where a higher number of dropped packets occurred.

This result allows the conclusion that the system is implemented and prepared to recover from a faulty or missing device in a dynamic self-repairing way, introducing very reduced amounts of information loss.

5.2 Experimental Tests

In this section, the results for repeating the tests described in Section 5.1 in the actual MTM-CM5000 motes will be under discussion. The methodology applied to the tests is the same as the one used in simulation environment: 30 minutes tests, when testing for variable rate, split the test in 10 minute periods like it was done in Section 5.1.2 and simulate a dangerous condition by altering the luminosity data by subjecting the photodiode sensor to a light source. These tests are of great importance since they will offer a real perspective about how the actual devices would behave out of the simulation environment.

Due to the devices range, the difficulty in setting some of the motes outside the range of the sink node, while still being in full control of the devices, did not allow to repeat the simulation test in Section 5.1.3 for the actual devices.

Upon testing the devices' functionalities during the development stages it was observed that the MTM-CM5000 devices had the capability to recover from a missing node. This was observed during the same preliminary test mentioned in Section 4.2.3, when the decision to include a periodic repair feature was made.

5.2.1 Experimental - Baseline Test

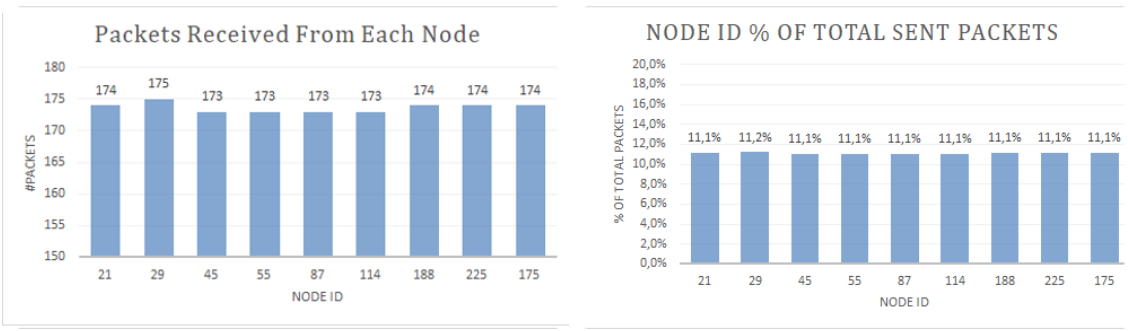
The objective of this test, like the baseline simulation test described in Section 5.1.1, is to determine the behaviour of the network when the code is deployed on real devices, like the MTM-CM5000. For this test, the same procedure from the simulation baseline tests were adopted and the expected number of received packets is maintained, for the same calculations apply to this test and 1620 packets are expected to be received for a 100% received packet ratio. After letting the system run for 30 minutes the obtained results were as shown in Table 5.6.

Table 5.6: Number of Expected and Received packets

# Expected Received Packets	# Packets Received	# Packets Received in Database	Ratio Received/Expected
1620	1563	1563	96.48%

By comparing this test to its analogous simulation environment test, it is clear that the results obtained are considerately worse with only 96.48% of the expected packets received against the simulation 98.64%, a 2.16% decrease in the packet delivery ratio. A drop in performance was to no surprise since the simulation environment is always expected to produce slightly different results from reality. This drop may be partly explained by small differences between the operation mode in a real mote and a simulated one. Physical limitations and conditions like real CPU performance, environment temperature, radio interferences, amongst other factors may impact the devices' performance. One operation aspect of the devices that may be affected by these factors, that may contribute greatly for this difference in results, are the differences in the actual timers regulating the reading rate. These timers are based on the microcontroller CPU tick rate, meaning that with impacted CPU performance comes different timer expirations. Another factor that may increase the deviation between expected and obtained results is the added inaccuracy in managing the start and stop time for the system test in real life.

Another important result is shown in Figure 5.11a and 5.11b, that shows that all devices worked in similar conditions and, share basically the same percentage of sent packets. Considering the obtained results, and comparing them to the ones obtained during the simulation test it is observable that the number of packets, on average, sent by each device is lower. This fact is translated in the difference between reading rates with the simulated devices, averaging 178 transmitted packets, against the averaged 174 transmitted packets with the real device test. This means that the MTM-CM5000 motes, with an expected reading rate of one transmitted packet every 10 seconds, for a 30 minute test, realized, on average, 4 reading cycles less when compared to the simulated motes.



(a) Total received packets / Node Id (b) Percentage of total received packets / Node Id

Figure 5.11: MTM-CM5000 tests obtained data.

It was once again tested and proved that all received data was successfully collected and stored in the database, as the results in Figure 5.12 also confirm.

NAME	DOCUMENTS	CAPPED?	SIZE
nodes	1,563	false	430.20 KB

Figure 5.12: Number of stored entries in the Database

5.2.2 Experimental - Variable Rate Test

For this test, the MTM-CM5000 motes were again utilized, and like in the analogous simulation for variable rate test, as described in Section 5.1.2, the purpose for this test was to observe the behaviour of the implemented alert mode feature, that upon being activated is translated in an increase of reading rates in each device. In order to realize this test and simulate a danger situation, since temperature and humidity variable changes are hard to emulate in real life, the luminosity value was used as a the danger index parameter.

The motes were all set for 10 minutes untouched and then a light source was used to raise the luminosity readings, in a selected sensor node, above a certain threshold in order to activate the alert mode for that mote. The procedure was repeated after the 20 minute mark for another mote, by adding it under the same light source. In order to limit the alerted motes, the motes set to alert mode, were in a different room division.

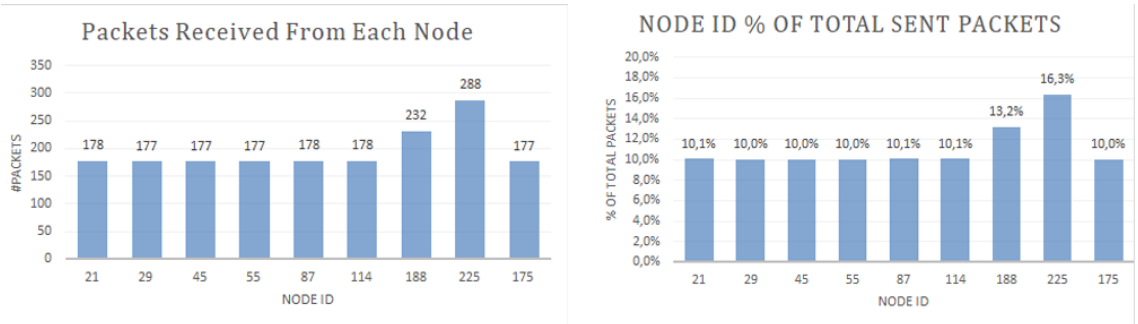
The results of this test will be compared against its analogous rate variation simulation test and the baseline device test. The calculations to determine the number of expected packets to be received are the same as in the variable rate simulation test, as described in 5.2.

After completing the test, the obtained results are as show in Table 5.7

Table 5.7: Number of Expected and Received packets

# Expected Received Packets	# Packets Received	# Packets Received in Database	Ratio Received/Expected
1800	1762	1762	97.89%

When analysing these results and comparing them to the previous baseline test where there was a considerable difference between the obtained results from the real device implementation and its analogous simulation tests, for this test the same does not apply. In fact, the results obtained in this test are very similar to the ones analysed in the rate variation simulation test. The difference in received packet ratio between the implementation of a variable rate feature in the MTM-CM5000 and the same simulated situation is only of 0.11%. Similarly to the simulation test, the nodes set on alert mode can easily be identified by looking at the results in Figures 5.13a and 5.13b.



(a) Total received packets / Node Id (b) Percentage of total received packets / Node Id

Figure 5.13: Obtained charts from simulation data

With a 16.3% share of all the transmitted packets, node 225 was the first node to be moved under the light source at the 10 minute mark time, entering the alert mode and increasing its reading rate. The second mote to be subjected to the same procedure, at the 20 minute mark, as can also be seen in the presented data, was mote 188 with a 13.2% share of transmitted packets. These collected statistics realistically represent the exact actions that took place during this test.

As can be observed in Figure 5.14, all the packets received at the gateway were also stored in the database, demonstrating once more the correct connection between the WSN and the web server.

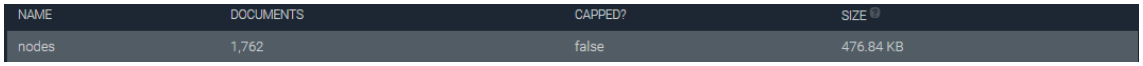


Figure 5.14: Number of stored entries in the Database

After this test and result analysis it was proved that the alert mode feature was successfully implemented and behaved as well in a real life tests as it did in the analogous simulation test. When compared to the previous baseline test it would be expected for the packet received ratio results to be lower then simulated ones but, actually, the system behaved almost exactly as designed with a deviation from the simulated results that can be considered almost non existent, less then 1%.

As a conclusion, after all the above tests were executed and the data was analysed, greater knowledge about the system and its features behaviour was obtained. It was also possible to determine that the previously proposed features for the system's network and its composing devices were successfully implemented and had the intended behaviour, both in simulation and experimental tests with MTM-CM5000 devices.

Chapter 6

Future Developments

In terms of possible future developments, considering the developed system, some improvements could be made. The main additions that could turn this system into a fully automatic, self-sustained system with a long operation life in deployments in remote areas and even more purpose oriented for early detection of forest fire occurrences would require the addition of extra hardware and some extra programming functions to the devices.

For this system to be deployed in real life, the first changes that would be required would be to switch the gateway responsible for connecting the sink node to the web application to a micro-computer like a Raspberry Pi with an integrated GSM/GPRS module to provide internet connection in areas without an internet infrastructure.

In order to minimize energy consumption, both time synchronization and sleep cycle management would be implemented on the sensor nodes and sink. Also, by changing or developing extra code for the devices, RPL objective function would be changed to consider not only the ETX metric, but also the alert state of a node. The preferred parent selection would then avoid an alerted sensor node, reducing the process and transmission loads for that node.

The next step would be to extend the life of the network by providing a power source to each sensor node, sink node and gateway. For this purpose, a photovoltaic panel would be integrated with the devices, removing the energy parameter from the list of variables that affect the system life time. Another solution would be to add a wind energy generator. This would complement the photovoltaic panel in energy production and provide extra information about the monitored site conditions, such as wind speed and direction.

Focusing on improving the developed web application, the alerts generated are only presented in the web client. To allow for more extensive alerting capabilities and reaching the system manager or owner at all times, SMS and Email messaging integration would be implemented. This development could also allow the system to directly alert the responsible authorities (Fire Corporations, Police, Civil Protection, etc.) upon detection of an ongoing fire situation.

Chapter 7

Conclusion

A fire detection oriented system capable of monitoring a large area, automatically recover from a missing or faulty device by applying the RPL routing protocol, with dynamic operation behaviour adjusted by environment conditions, that is able to present and process the monitored environment data in real time and produce alerts was intended for development during this project.

Considering all the stated development stages, the realized tests and collected statistics it can be concluded that this project fulfilled the implementation of the targeted features, although leaving room for some possible future improvements in orienting the system even more towards the fire detection paradigm and expanding or changing the hardware in order to create a more robust and self sustainable deployable wireless sensor network.

After several tests and result analysis, all the intended and implemented mechanisms were proven to function as expected and the project was concluded with its main objectives fulfilled.

References

[1] Silicon Labs. The Evolution of Wireless Sensor Networks. *White Papers*, pages 1–5, 2013. URL: http://www.silabs.com/SupportDocuments/TechnicalDocs/evolution-of-wireless-sensor-networks.pdf. Cited on pages vii, 3, 4, and 5.

[2] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless Sensor Networks*. 2007. URL: http://www.amazon.com/Wireless-Sensor-Networks-Technology-Applications/dp/0471743003$\delimiter"026E30F$nhttp://doi.wiley.com/10.1002/047011276X, doi:10.1002/047011276X. Cited on pages vii, 10, 11, and 14.

[3] Ieee Standard and Ieee Computer Society. *Local and metropolitan area networks — Part 15 . 4 : Low-Rate Wireless Personal Area Networks (LR-WPANs) IEEE Computer Society*, volume 2011. 2011. Cited on pages vii, 13, 14, 15, and 43.

[4] Elaheh Rahmani. Presented by : Elaheh Rahmani May 2005 Outline :. (May), 2005. Cited on pages vii and 14.

[5] Fabian Nack. An overview on wireless sensor networks. *Institute of Computer Science (ICS) University, Barlin*, 2010. Cited on pages vii, 3, 8, and 16.

[6] Wenxiang Li, Yunhe Wu, Chunsheng Zhu, and Yajie Ma. Performance Comparison of Source Routing Tactics for WSN of Grid Topology. 12:3–8, 2014. doi:10.1109/DASC.2014.59. Cited on pages vii, 16, 17, 18, 19, and 20.

[7] Crossbow Technology Inc. TelosB Mote Platform. pages 1–2, 2004. doi:6020-0094-01Rev.B. Cited on pages vii, 26, and 27.

[8] Memsic. MICAz datasheet: 6020-0065-05 rev. *San Jose, CA, California*, Revision 6:1–2, 2003. Cited on pages vii, 27, and 28.

[9] Libelium Comunicaciones Distribuidas S.L. Waspmote Datasheet, 2016. Cited on pages vii, 32, 33, and 34.

[10] Tiago Couto Braga. Monitorização ambiental em espaços florestais com rede de sensores sem fios. 2010. URL: http://digituma.uma.pt/handle/10400.13/229. Cited on pages vii, 3, 5, 8, 10, 12, 13, 16, 37, 38, 39, and 41.

[11] J.T. Jon T Adams, An Introduction, and J.T. Jon T Adams. An introduction to IEEE STD 802.15.4. *2006 IEEE Aerospace Conference*, pages 1–8, 2006. doi:10.1109/AERO.2006.1655947. Cited on pages vii and 43.

[12] S Gowrishankar, T G Basavaraju, D H Manjaiah, and Subir Kumar Sarkar. Issues in Wireless Sensor Networks. *Proceedings of the World Congress on Engineering 2008 Vol I*, I:978–988, 2008. Cited on pages ix, 5, 6, and 10.

[13] Chee-Yee Chong and S. P. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug 2003. doi:10.1109/JPROC.2003.814918. Cited on pages ix, 3, 4, 5, and 6.

[14] Tiago Camilo, André Rodrigues, Jorge Sá Silva, and Fernando Boavida. Redes de Sensores Sem Fios , considerações sobre a sua instalação em ambiente real. (May 2016). Cited on pages 8, 16, 23, and 24.

[15] Faisal Bashir Hussain and Jae-young Pyun. Coordinator Discovery and Association in Beacon-Enabled IEEE 802 . 15 . 4 Network. 2013, 2013. Cited on pages 12 and 13.

[16] Emanuele Toscano, Lucia Lo Bello, and Viale A Doria. low-power industrial WSNs in realistic scenarios. 2012. Cited on page 15.

[17] Samir R. Das, Charles E. Perkins, and Elizabeth M. Belding-Royer. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July 2003. URL: https://rfc-editor.org/rfc/rfc3561.txt, doi:10.17487/RFC3561. Cited on page 20.

[18] Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, March 2012. URL: https://rfc-editor.org/rfc/rfc6550.txt, doi:10.17487/RFC6550. Cited on page 21.

[19] Hanane Lamaazi, Nabil Benamar, and Antonio J. Jara. Rpl-based networks in static and mobile environment: A performance assessment analysis. *Journal of King Saud University - Computer and Information Sciences*, pages –, 2017. URL: http://www.sciencedirect.com/science/article/pii/S1319157816301574, doi:https://doi.org/10.1016/j.jksuci.2017.04.001. Cited on page 21.

[20] Torsten Braun. iCIS Summer Workshop , June 24-25 , 2014 Coimbra I . Internet of Things. 2014. Cited on page 21.

[21] Volkan Isler and Sampath Kannan. Sampling based sensor-network deployment. *Proceedings of 2004 1EEEiRS.J International Conference on Intelligent Robots and Systems*, (January):1780–1785, 2004. URL: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1389654, doi:10.1109/IROS.2004.1389654. Cited on page 23.

[22] Sasa SlijepcevicandMiodrag Potkonjak. Power efficient organization of wireless sensor networks. *Proceedings of the IEEE International Conference on Communications 2001*, pages 472–476, 2001. Cited on page 23.

[23] Libelium Comunicaciones Distribuidas S.L. Libelium Comunicaciones Distribuidas S.L. http://www.libelium.com, 2016. [Online; accessed 10-July-2016]. Cited on page 31.

[24] Libelium Comunicaciones Distribuidas S.L. Libelium Comunicaciones Distribuidas S.L. http://www.libelium.com/wireless_sensor_networks_to_detec_forest_fires/, 2016. [Online; accessed 10-July-2016]. Cited on page 31.

[25] Libelium Comunicaciones Distribuidas S.L. Meshlium Datasheet, 2014. Cited on page 32.

[26] Byungrak Son, Yong-sork Her, and J Kim. A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains. *International Journal of Computer Science and Network Security (IJCSNS)*, 6(9):124–130, 2006. Cited on page 35.

[27] Sensorion. SHT1x / SHT7x Humidity / Temperature Sensor. (September):1–11, 2008. Cited on pages 43 and 49.

[28] Hamamatsu. Si photodiodes S1087/S1133 series . pages 3–7. Cited on pages 43 and 49.

[29] Jürgen Schönwälder. Internet of Things. *Jucs*, 2010. doi:10.1016/j.clsr.2009.11.008. Cited on page 43.

[30] Carla Willis, Brian Van Wilgen, Kevin Tolhurst, Colin Everson, Peter D'Abreton, Lionel Pero, and Gavin Fleming. The Development of a National Fire Danger Rating System for South Africa. (July):1–89, 2001. URL: http://www.daff.gov.za/doaDev/sideMenu/ForestryWeb/dwaf/cmsdocs/Elsa/Docs/Fire/DevofNatFireDangerRatingSystem2001.pdf. Cited on page 50.

[31] Marcos Leonardo Lopes Magalhães. *Sensorização Ambiente : Indicadores de Conforto*. PhD thesis, 2015. Cited on page 51.